

Final Report

January 1975

Finite Element Analysis of Mercury Slosh in the Solar Electric Propulsion Stage

(NASA-CR-120597) FINITE ELEMENT ANALYSIS OF
MERCURY SLOSH IN THE SOLAR ELECTRIC
PROPULSION STAGE Final Report (Martin
Marietta Corp.) 136 p HC \$5.75

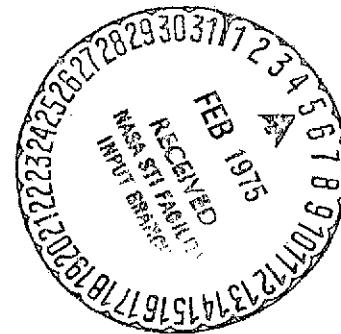
N75-15737

CSCL 21C

Unclass

G3/20

08547



Prepared for:
Marshall Space Flight Center
Huntsville, Alabama 35823

MARTIN MARIETTA

MCR-74-464

"FINITE ELEMENT ANALYSIS OF MERCURY SLOSH
IN THE SOLAR ELECTRIC PROPULSION STAGE"

Contract NAS8-29944

(Final Report)

Approved by:


George Morosow
George Morosow
Program Manager

January 1975

Author: J. N. Singh

Prepared for: Marshall Space Flight Center
Huntsville, Alabama 35812

MARTIN MARIETTA CORPORATION
Denver Division
Denver, Colorado 80201

FOREWORD

This report, prepared by the Dynamics and Loads Section, Martin Marietta Corporation, Denver Division, under Contract NAS8-29944, presents the technical approach and the results of a study contract for the dynamic characteristics of a spherical tank/fluid/bladder to be used in Solar Electric Propulsion Stage (SEPS). The study was administered by the National Aeronautics and Space Administrations, George C. Marshall Space Flight Center, Huntsville, Alabama, under the direction of Mr. Frank Bugg, Systems Dynamics Laboratory.

CONTENTS

	<u>Page</u>
Introduction	1
1. Scope	1
1.1 Purpose	2
1.2 Scope	2
1.3 Summary	2
2. Technical Approach	2
2.1 Static Equilibrium Shape	3 thru 11
2.2 Vibration Analysis	12
2.2.1 Rayleigh-Ritz Technique	12 thru 17
2.2.2 Mass and Stiffness Matrices	17 thru 20
2.2.3 Stiffness and Mass Matrices in Local Coordinates (Solid)	21 thru 22
2.2.4 Stiffness and Mass Matrices in Local Coordinates (Fluid)	22 thru 24
2.3 Mechanical Equivalent	24 thru 26
3. Model	27
3.1 Finite Element Model of the System	27 thru 28
3.2 Reflection of Quadrants 2, 3 and 4 on Quadrant 1	28 thru 31
4. Results	32
4.1 Static Equilibrium Shape	32 thru 46
4.2 Vibration Analysis	47
4.3 Mechanical Equivalent	47 thru 53
5. Conclusions and Comments	54 thru 55
6. References and Bibliography	56 thru 58
7. Appendices	59 thru 61
A1 - Static Free Surface Program	A1-1 thru A1-20
A2 - Dynamic Analysis Program	A2-1 thru A2-43
B1 - Explanation of Input to Static Free Surface Program	B1-1 thru B1-2

B2 - Explanation of Input to Dynamic Analysis Program	B2-1 thru B2-6
---	-------------------

Figures

1G Static Free Surface for 1g, 80% Full	33
2G Static Free Surface for 1g, 60% Full	34
3G Static Free Surface for 1g, 50% Full	35
4G Static Free Surface for 1g, 40% Full	36
5G Static Free Surface for 1g, 20% Full	37
6G Static Free Surface for $10^{-5}g$, 80% Full	38
7G Static Free Surface for $10^{-5}g$, 60% Full	39
8G Static Free Surface for $10^{-5}g$, 50% Full	40
9G Static Free Surface for $10^{-5}g$, 40% Full	41
10G Static Free Surface for $10^{-5}g$, 20% Full	42
11G Polynomial Coefficient A_1 vs Percentage Full	43
12G Polynomial Coefficient A_3 vs Percentage Full	44
13G Polynomial Coefficient A_4 vs Percentage Full	45
14G Polynomial Coefficient A_5 vs Percentage Full	46

ACKNOWLEDGEMENTS

The author appreciates the assistance necessary for the successful completion of the study. Mr. Frank Bugg, of NASA, had many valuable comments and suggestions. The very helpful comments and suggestions provided by Mr. Carl Bodley of the Dynamics and Loads Section, Martin Marietta Corporation, Denver Division, is very deeply appreciated. Thanks are also due to Mr. Peter Abbott, Mr. Richard Wohlen, Mr. Dave Warner, Mr. Wilcomb Benfield, Mr. Roger Philippus, for their effort and assistance. The author also is very deeply appreciative of the helpful guidance provided by Mr. George Morosow, the Program Manager.

INTRODUCTION

The proposed Solar Electric Propulsion Stage (SEPS) will be responsible for the transmittal of Shuttle-based payloads between low orbit and geosynchronous orbit. The SEPS will be a reusable vehicle powered by accelerated mercury ions. The mercury propellant will be contained in spherical tanks. Propellant orientation within the tank will be controlled by a neoprene expulsion bladder; propellant feed to the ion engines will be maintained by insertion of freon pressurant into the ullage space between the tank and the expulsion bladder. The mass of the mercury will be a very significant portion of the total mass of the spacecraft. It is, therefore, apparent that this high mass fraction cannot be ignored and its dynamic characteristics must be investigated.

This study obtained the equilibrium shapes, vibration modal properties, and mechanical equivalent slosh models for the mercury and bladder for five fill conditions and two acceleration levels.

1. SCOPE

1.1 Purpose

The purpose of this report is to document the investigation performed under contract NAS8-29944, "Finite Element Analysis of Mercury Slosh in the Solar Electric Propulsion Stage".

1.2 Scope

This report documents the dynamic characteristics of the system of spherical tank/hemispherical expulsion bladder/mercury propellant. The static equilibrium shape corresponding to various ullage and gravity configurations was established. The finite-element approach was applied to different fill conditions in different gravity fields to evaluate the lateral sloshing mode shapes and frequencies. The resulting mode shapes and frequencies was used to define a spring-mass mechanical analog that describes the sloshing phenomenon. Computer programs for the equilibrium shape and vibration analysis including the modeling were developed.

1.3 Summary

The static equilibrium shapes of the neoprene bladder have been established corresponding to various ullage and gravity configurations under specified boundary conditions. The hemispherical bladder is taken to be attached at the diametral plane of the sphere with zero relative slope. With these shapes, the spherical tank with bladder and mercury has been modeled as an assemblage of finite-elements. The properties of these elements have then been calculated using a linear displacement field. The dynamic characteristics were obtained to be used to define a mechanical analog which will reproduce the sloshing phenomenon of the system. The computer programs for the static free surface and vibration analysis have been checked out.

2. TECHNICAL APPROACH

The problem was approached in two distinct steps.

- (1) Establish the static equilibrium shape.
- (2) Using the free surface shape from (1), model the system as an assemblage of finite elements and obtain the mode shapes and frequencies. Further define a mechanical analog for the system.

2.1 Static Equilibrium Shape - The method comprises of determining the total energy of the system in terms of the variables involved and minimizing the total energy for the stable equilibrium state under the given constraints. This is outlined below. However, it is to be pointed out here that the idealized case here is for an infinitely long cylinder and the feeling being that the static free surface for this case will be close to the actual case of spherical tank.

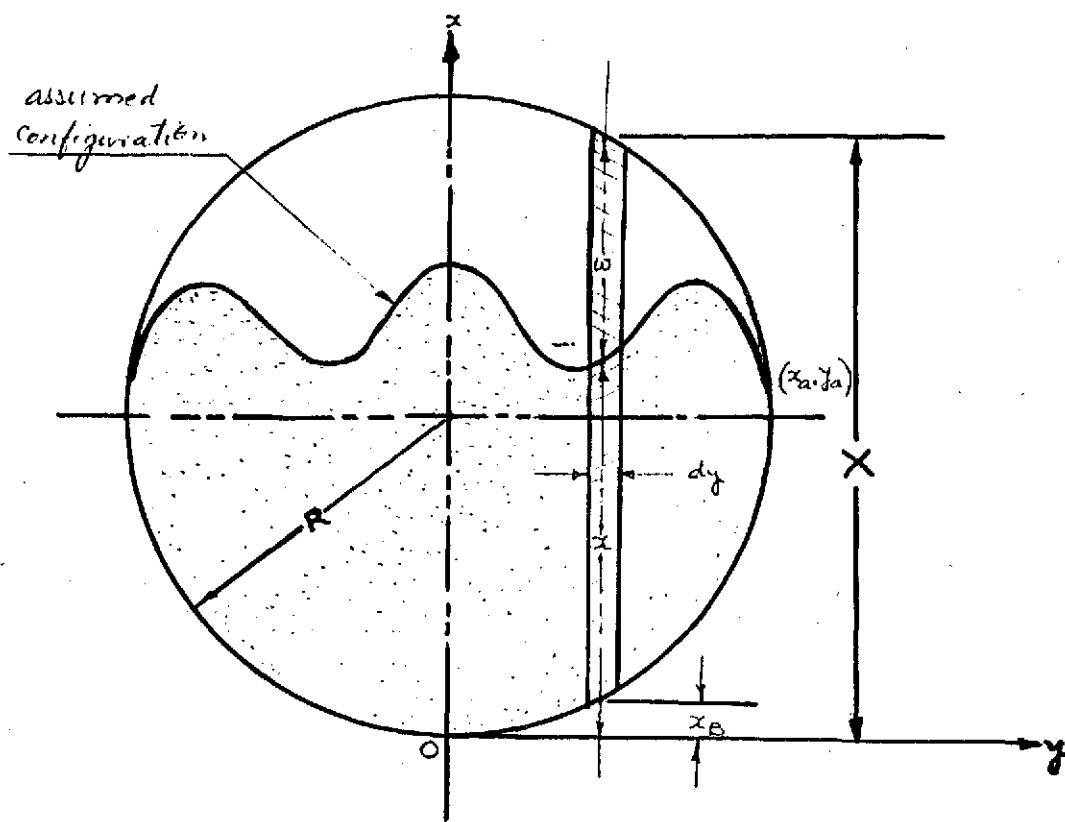


Figure 1

The equation for the circle (with origin at center)

$$\bar{x}^2 + \bar{y}^2 = R^2 \quad (1)$$

Taking the origin at bottom point

$$x = \pm [R^2 - y^2]^{1/2} + R \quad (2)$$

Introducing $w(y)$ in the expression

$$x(\gamma) = x(\gamma) - w(\gamma) = \pm [R^2 - \gamma^2]^{1/2} + R - w(\gamma) \quad (4)$$

The plus sign represents the upper point of the sphere and the negative sign the bottom and hence,

$$x(\gamma) = x(\gamma) - w(\gamma) = [R^2 - \gamma^2]^{1/2} + R - w(\gamma) \quad (5)$$

An assumption is made here that $w(y)$ may be expressed in terms of a power series:

$$w(\gamma) = \sum_{i=0}^N a_i \gamma^i \quad (6)$$

Substituting for $w(y)$ from equation (6) in equation (5),

$$x(\gamma) = [R^2 - \gamma^2]^{1/2} + R - \sum_{i=0}^N a_i \gamma^i \quad (7)$$

The stretching in the membrane strip of length s measured along the curved shape is $\tilde{u}(s)$. It is assumed that

$$\tilde{u}(s) = 5s \quad (8)$$

Therefore,

$$\frac{\partial \tilde{u}}{\partial s} = 5 \quad (9)$$

That is, 5 is the strain in the membrane.

The total potential of the assumed configuration will now be calculated. They are:

(i) Bladder strain energy in bending:

$$dV_B = \frac{1}{2} \frac{EI ds}{\rho^2} \quad (10)$$

where: dV_B = strain energy in bending

E = modulus of elasticity for the bladder

I = moment of inertia
 ds = length
 ρ = radius of curvature

Further

$$\frac{1}{\rho} = \frac{\frac{d^2x/dy^2}{2}}{\left[1 + \left(\frac{dx}{dy}\right)^2\right]^{3/2}} \quad (11)$$

Therefore, equation (10) can be rewritten as

$$dV_B = \frac{1}{2} EI \frac{\left(\frac{d^2x/dy^2}{2}\right)^2}{\left[1 + \left(\frac{dx}{dy}\right)^2\right]^3} ds \quad (12)$$

It can easily be seen that

$$ds = \left[\frac{1}{1 + \left(\frac{dx}{dy}\right)^2} \right]^{-1/2} dy \quad (13)$$

Substituting from (13) in (12)

$$dV_B = \frac{1}{2} EI \frac{\left(\frac{d^2x/dy^2}{2}\right)^2}{\left[1 + \left(\frac{dx}{dy}\right)^2\right]^5} s^{1/2} dy \quad (14)$$

Therefore,

$$V_B = \frac{1}{24} Eh^3 \int_0^{ya} \frac{\left(\frac{d^2x/dy^2}{2}\right)^2}{\left[1 + \left(\frac{dx}{dy}\right)^2\right]^5} dy \quad (15)$$

where: h = thickness of the bladder

(ii) The strain energy due to membrane action:

$$dV_M = \frac{1}{2} (\text{stress}) (\text{strain}) h ds$$

Resulting in

$$V_M = \frac{1}{2} Eh^2 \int_0^{ya} \left[1 + \left(\frac{dx}{dy}\right)^2\right]^{1/2} dy \quad (16)$$

where V_M = strain energy due to membrane action.

(iii) Fluid gravitational potential energy:

$$dV_g = (x - x_B) dy \cdot 1 \cdot \rho \cdot g \cdot \{x_B + (x - x_B)/z\} \quad (17)$$

Simplifying,

$$V_g = \frac{1}{2} \rho g \int_0^{y_a} x^2 dy - \frac{1}{2} \rho g \int_0^{y_a} x_B^2 dy \quad (18)$$

where: ρ = mass density of the fluid

g = acceleration due to gravity

x_B = as shown in Figure 1

V_g = gravitational potential energy

(iv) Work done by ullage pressure:

$$dW = p \cdot dV \quad (19)$$

where: p = ullage pressure

Therefore, virtual work by ullage pressure due to virtual change in w ,

$$\delta W = \oint_{\gamma_a} \delta w \cdot dy \quad (20)$$

where: δw = virtual change w

δW = virtual work

Substituting for δw from (6) in (20),

$$\delta W = \oint \sum_{i=0}^N \delta a_i \int_0^{y_a} y^i dy \quad (21)$$

as

$$w = w(a_0, a_1, a_2, \dots, a_N, \delta) \quad (22)$$

$$\delta w = \sum_{i=0}^N \frac{\partial w}{\partial a_i} \delta a_i + \frac{\partial w}{\partial \delta} \delta \delta \quad (23)$$

The comparison of equations (21) and (23) reveals,

$$\frac{\partial w}{\partial a_i} = \rho \int_0^{y_a} y^i dy \quad (24)$$

for $i = 0, 1, 2, \dots, N$

and

$$\frac{\partial w}{\partial b} = 0 \quad (25)$$

In the present case, the system may be treated as conservative for a particular time, and therefore

$$V = -W \quad (26)$$

where: V = is the total potential energy of the system

W = work function

Substituting in equation (24),

$$\frac{\partial V}{\partial a_i} = -\rho \int_0^{y_a} y^i dy \quad (27a)$$

and

$$\frac{\partial V}{\partial b} = 0 \quad (27b)$$

under the boundary conditions of

$$\begin{aligned} \omega(y_a) &= 0 \\ \omega'(y_a) &= 0 \\ \omega'(0) &= 0 \\ \int_0^{y_a} \omega dy &= k \end{aligned} \quad (28)$$

where: $2K =$ given ullage volume

and

$$V = V_B + V_M + V_g \quad (29)$$

Assembling the expression

$$V = \frac{\rho h^3}{24} \int_0^{y_a} \frac{\frac{d^2}{dy^2}}{\left[1 + \left(\frac{dx}{dy}\right)^2\right]^{5/2}} dy + \frac{\rho h E^2}{2} \int_0^{y_a} \left[1 + \left(\frac{dx}{dy}\right)^2\right]^{1/2} dy + \frac{\rho g}{2} \int_0^{y_a} (x^2 - x_B^2) dy \quad (30)$$

The equation (30) has to be expressed in terms of a_i 's and

$$x(\gamma) = X(\gamma) - \omega(\gamma) = (R^2 - \gamma^2)^{1/2} + R - \sum_{i=0}^N a_i \gamma^i \quad (31)$$

$$x' = \frac{dx}{d\gamma} = -\gamma (R^2 - \gamma^2)^{-1/2} - \sum_{i=0}^N i a_i \gamma^{i-1} \quad (32)$$

$$x'' = \frac{d^2x}{d\gamma^2} = - (R^2 - \gamma^2)^{-1/2} - \gamma^2 (R^2 - \gamma^2)^{-3/2} - \sum_{i=0}^N i(i-1) a_i \gamma^{i-2} \quad (33)$$

Substituting from (31), (32) and (33) in equation (30) and differentiating with respect to a_i 's and b respectively and comparing with (27a) and (27b),

$$\begin{aligned} \frac{\partial V}{\partial a_i} &= -\rho \int_0^{\gamma_a} y^i dy \\ &= \frac{Eh^3}{12} \int_0^{\gamma_a} \left[\{1+(x')^2\} \{x''\} \{-i(i-1)y^{i-2}\} - \left\{ \frac{h^2}{2} \right\} \{x''\}^2 \{x\} \{-iy^{i-1}\} \right] dy \\ &\quad + \frac{Eh^2}{2} \int_0^{\gamma_a} \left[\{1+(x')^2\}^{1/2} \{x'\} \{-i y^{i-1}\} \right] dy \\ &\quad + \rho g \int_0^{\gamma_a} [\{x\} \{-y^i\}] dy \end{aligned} \quad (34a)$$

and

$$\frac{\partial V}{\partial b} = 0 = E h \int_0^{\gamma_a} [1 + \{x'\}^2] dy \quad (34b)$$

for $i = 0, 1, 2, \dots, N$

Under the auxiliary conditions of

$$\begin{aligned} \omega(\gamma_a) &= 0 \\ \omega'(\gamma_a) &= 0 \\ \omega'(0) &= 0 \\ \int_0^{\gamma_a} \omega dy &= K \end{aligned} \quad (35)$$

Finally, the equations of the system are

$$\begin{aligned} & \frac{Eh^3}{12} \int_0^{\gamma_a} \left[\{1+(x')^2\} \{x''\} \{-L(x-1) y^{i-2} - \frac{1}{2} \{x'\} \{x''\}^2 \{x'\} \{-i y^{i-1}\} \right] dy \\ & + \frac{Eh^5}{2} \int_0^{\gamma_a} \left[\{1+(x')^2\}^{-\frac{1}{2}} \{x'\} \{-i y^{i-1}\} \right] dy \\ & + \rho g \int_0^{\gamma_a} [\{x\} \{-y^i\}] dy + \rho \int_0^{\gamma_a} y^i dy = 0 \end{aligned} \quad (36)$$

and

$$Eh^5 \int_0^{\gamma_a} [1 + \{x'\}^2] dy = 0 \quad (37)$$

for $i = 0, 1, 2, \dots, N$

under the conditions

$$\begin{aligned} w(\gamma_a) &= 0 \\ w'(\gamma_a) &= 0 \\ w'(0) &= 0 \\ \int_0^{\gamma_a} w dy &= k \end{aligned} \quad (38)$$

Thus,

$$F = F(a_0, a_1, \dots, a_N, b) = F(q_1, q_2, \dots, q_{N+2}) \quad (39)$$

In general, the equations can be written as,

$$\begin{aligned} F_n(q_i) &= \frac{Eh^3}{12} \int_0^{\gamma_a} \left[\{1+(x')^2\} \{x''\} \{-n(n-1)(n-2) y^{n-3} - \frac{1}{2} \{x'\} \{x''\}^2 \{x'\} \{-(n-1) y^{n-2}\} \right] dy \\ & + \frac{Eh^5}{2} \int_0^{\gamma_a} \left[\{1+(x')^2\}^{-\frac{1}{2}} \{x'\} \{-(n-1) y^{n-2}\} \right] dy \\ & + \rho g \int_0^{\gamma_a} [\{x\} \{-y^{n-1}\}] dy + \rho \int_0^{\gamma_a} y^{n-1} dy = 0 \end{aligned} \quad (40a)$$

for $n = 1, 2, \dots, N+1$

$$F_n(q_i) = Eh^5 \int_0^{\gamma_a} [1 + (x')^2]^{\frac{1}{2}} dy = 0 \quad (40b)$$

for $n = N+2$

under the auxiliary conditions

$$\begin{aligned} w(\gamma_a) &= 0 \\ w'(\gamma_a) &= 0 \\ w'(0) &= 0 \\ \int_0^{\gamma_a} w dy &= k \end{aligned} \quad (40c)$$

The system of equations (40), i.e., (40a), (40b) are to be solved under conditions of (40c) for q_i with iterative approach. Denoting $(F_n)_s$, where s stands for iteration step, $(F_n)_0$ stands for initial assumption,

$$(F_n)_1 = (F_n)_0 + \frac{\partial}{\partial q_m} (F_n)_0 \Delta q_m \quad (41)$$

In general, therefore,

$$(F_n)_{s+1} = (F_n)_s + \frac{\partial}{\partial q_m} (F_n)_s \Delta q_m \quad (42)$$

where $n = 1, 2, \dots, N+2$ and $m = 1, 2, \dots, N+2$ (separately).

Rewriting the equations (42) in matrix form:

$$\left\{ \begin{array}{l} F_1 \\ F_2 \\ F_3 \\ \vdots \\ F_{N+2} \end{array} \right\}_{s+1} = \left[\begin{array}{ccc|c|c} \frac{\partial F_1}{\partial q_1} & \frac{\partial F_1}{\partial q_2} & \frac{\partial F_1}{\partial q_3} & \cdots & \frac{\partial F_1}{\partial q_{N+2}} \\ \frac{\partial F_2}{\partial q_1} & \frac{\partial F_2}{\partial q_2} & \frac{\partial F_2}{\partial q_3} & \cdots & \frac{\partial F_2}{\partial q_{N+2}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_{N+2}}{\partial q_1} & \frac{\partial F_{N+2}}{\partial q_2} & \frac{\partial F_{N+2}}{\partial q_3} & \cdots & \frac{\partial F_{N+2}}{\partial q_{N+2}} \end{array} \right] \left\{ \begin{array}{l} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \vdots \\ \Delta q_{N+2} \end{array} \right\} \quad (43)$$

The various terms involved in the matrix are as follows:

$$\begin{aligned} \frac{\partial F_n}{\partial q_m} &= \frac{\epsilon A^3}{12} \int_0^{\gamma_a} \left(\left(1 + (x')^2 \right)^{-\frac{3}{2}} \right) \left(\left[\left\{ (n-1)(n-2) \right\} x'^{n-3} \right] \left[\left\{ 1 + (x')^2 \right\} \frac{\partial}{\partial q_m} (x') \right. \right. \\ &\quad \left. \left. + (x'') \left\{ 2(x') \frac{\partial}{\partial q_m} (x') \right\} \right] - \left(\frac{1}{2} \right) \left[-(n-1) \right] x'^{n-2} \left\{ (x')^2 \frac{\partial}{\partial q_m} (x') + 2(x')(x'') \frac{\partial}{\partial q_m} (x'') \right\} \right) \\ &\quad + \left([1 + (x')^2] [x''] [(n-1)(n-2) x'^{n-3}] - \left(\frac{1}{2} \right) (x'')^2 (x') [-(n-1) x'^{n-2}] \right) \quad (44a) \end{aligned}$$

$$\left(\left[-\frac{1}{2} \right] \left[1 + (x')^2 \right]^{-\frac{1}{2}} \left[z(x) \right] \left[\frac{\partial}{\partial a_m}(x') \right] \right] dy \\ + \frac{EhB^2}{2} \int_0^{y_a} \left[\left\{ 1 + (x')^2 \right\}^{-\frac{1}{2}} \left\{ \frac{\partial}{\partial a_m}(x) \right\} + (-\frac{1}{2}) \left\{ 1 + (x')^2 \right\}^{-\frac{1}{2}} \left\{ z(x) \frac{\partial}{\partial a_m}(x') \right\} \{x'\} \right] \\ (-\{n-1\} y^{n-2}) \] dy + \rho g \int_0^{y_a} \left(\left[\frac{\partial}{\partial a_m}(x) \right] [-y^{n-1}] \right) dy$$

for $n = 1, 2, \dots, N+1$ and $m = 1, 2, \dots, N+1$ (separately)

and

$$\frac{\partial F_n}{\partial q_m} = EhB \int_0^{y_a} \left(\left\{ 1 + (x')^2 \right\}^{-\frac{1}{2}} \{x'\} \{-\{n-1\} y^{n-2}\} \right) dy \quad (44b)$$

for $n = 1, 2, \dots, N+1$ and $m = N+2$ (separately)

Also

$$\frac{\partial F_n}{\partial q_m} = EhB \int_0^{y_a} \left(\left\{ \frac{1}{2} \right\} \left\{ 1 + (x')^2 \right\}^{-\frac{1}{2}} \left\{ z(x) \frac{\partial}{\partial a_m}(x') \right\} \right) dy \quad (44c)$$

for $n = N+2$ and $m = 1, 2, \dots, N+1$

and

$$\frac{\partial F_n}{\partial q_m} = Eh \int_0^{y_a} \left(1 + \{x'\}^2 \right)^{\frac{1}{2}} dy \quad (44d)$$

for $n = N+2$ and $m = N+2$

Recalling equation (13), the arc length can be written as

$$s = \int_0^{y_a} \left(\frac{1.0}{1 + (x')^2} \right)^{\frac{1}{2}} dy \quad (45)$$

Assuming the bladder to be attached at the diametral plane, the original length of the arc is

$$l = \pi R \quad (46)$$

Therefore, b can, now, be defined as

$$b = \frac{2s - l}{l} \quad (47)$$

Thus, the system of equations (40a), (40b), (40c) and (47) are to be solved simultaneously with an iterative approach. This is accomplished in the computer program.

2.2 Vibration Analysis - Vibration analysis of structures having a large number of degrees of freedom is an ever present problem. Digital computer oriented techniques are primarily restricted by core size and computer time. Consequently, economically feasible eigenvalue/eigenvector techniques are needed for large size structural systems.

However, for a large size structural system, if one is interested in a relatively small number of modes, Rayleigh-Ritz technique appears to have the advantage over the others. In this method, essentially a large problem is reduced to a smaller problem for the range of interest only. The solution is obtained as a linear combination of assumed linearly independent mode shapes.

The method consists in using the assumed mode shapes initially, which reduce the number of generalized coordinates used and then, by an iterative procedure, these modes are improved until they converge to the normal vibration modes of the structure.

2.2.1 Rayleigh-Ritz Technique - The iterative Rayleigh-Ritz method(1) is used to calculate the mode shapes and frequencies of the system. This method is based on repeated application of the well known Rayleigh-Ritz technique using improved mode shapes for each iteration. This technique reduces the size of the system without degrading accuracy in the desired frequency range. The technique is briefly described here.

For a discrete coordinate model of a structure having n degrees of freedom, the equations of motion can be written as

$$[M]\{\ddot{x}\} + [K]\{x\} = \{0\} \quad (48)$$

where:

$\{x\} = \{x(t)\}$ vector of discrete coordinate displacements,

$[M]$ = mass matrix

$[K]$ = stiffness matrix

If a solution of the type $\{x\} = \{x\} e^{i\omega t}$, implying a simple harmonic motion is assumed, equations (48) can be written as

$$([\kappa] - \omega^2 [M]) \{x\} = \{0\} \quad (49)$$

Equation (49) is recognized as a matrix eigenvalue problem of order n , whose eigenvectors $\{y\}$ are the mode shapes and whose eigenvalues $[\omega^2]$ are the frequencies. A complete sequence of trial vectors

$$\{z\}_1, \{z\}_2, \{z\}_3, \dots, \{z\}_n \quad (50)$$

which are linearly independent, is assumed. The displacement $\{x\}$ is then expressed as a linear sum of the first "m" trial vector, that is,

$$\begin{matrix} \{x\} = [z] \{q\} \\ (n \times 1) \quad (n \times m) \quad (m \times 1) \end{matrix} \quad (51)$$

Substitution of equation (51) into (49) and for multiplying by $[z]^T$ gives

$$([\bar{\kappa}] - \omega^2 [\bar{M}]) \{q\} = \{0\} \quad (52)$$

where:

$$[\bar{\kappa}] = [z]^T [\kappa] [z] \quad (53)$$

and

$$[\bar{M}] = [z]^T [M] [z] \quad (54)$$

Equation (52) is a matrix eigenvalue problem of reduced order "m" whose eigenvectors are $[Y^*]$ and eigenvalues are $[\omega^2]$. The solution of equation (52) has the form

$$\{q\} = [Y^*] \{q^*\} \quad (55)$$

where: $\{q^*\}$ is the normalized coordinate vector. The eigenvalues, $[\omega^2]$, approximate the first "m" eigenvalues of the original structure. The associated approximate eigenvectors $[Y]$ of the original structure are obtained by substitution of equation (55) into (51), yielding

$$\{x\} = [\mathbf{z}] [\mathbf{y}^*] \{q_r^*\}$$

(nx1) (nxm) (mxm) (mx1)

or

$$\{x\} = [\mathbf{y}] \{q_r^*\} \quad (56)$$

where: $[\mathbf{y}] = [\mathbf{z}] [\mathbf{y}^*]$ (57)

The accuracy of the mode shapes $[\mathbf{y}]$ and frequencies $[\omega^2]$ obtained depends entirely upon the trial vector $[\mathbf{z}]$. If $[\mathbf{z}]$ contains the true modal patterns, then the eigensolution for $[\mathbf{y}]$ and $[\omega^2]$ are exact. However, in general, that is not the case. Exact results can be obtained for the first "m" modes of the structure if the trial vectors $[\mathbf{z}]$ do not have any contribution from modes higher than "m". Thus, an improved set of trial vectors can be calculated by suppressing the contribution of higher modes in approximate mode shapes. The procedure for suppressing the contribution of the higher modes is well known; in fact, it is the basis of the Power or Stodola-Vianello¹ matrix iteration method⁽²⁾ of modal analysis. Here, however, the method is applied to all modes simultaneously and is given as,

$$[\mathbf{K}] [\mathbf{z}] = [\mathbf{M}] [\mathbf{y}] \quad (58)$$

The solution is carried out for $[\mathbf{z}]$, which is then used to repeat equations (52) through (58). The cycle can be repeated until all the mode shapes $[\mathbf{y}]$ and frequencies ω^2 have converged to within a prescribed tolerance. Convergence is assured because the technique is equivalent to a power iteration applied simultaneously to all modes. Thus, the convergence theorems associated with the power method are directly applicable. The role of the eigensolution (equation (52)) is to prevent all modes from converging on the lowest mode.

Associated with the iterative Rayleigh-Ritz technique are parameters that affect the convergence and hence computer time which will be briefly discussed here. They are:

- (i) the initial mode shapes assumed to start the iteration process,
- (ii) the number of modes used,
- (iii) the repression of higher modes, and
- (iv) shifting.

(i) Initially Assumed Mode Shapes - The choice of initial mode shapes plays a very important role in the success of the technique. Inherent with the initial mode shape selection are two basic problems: (1) modes may be missed, and (2) the triple product $[\bar{M}] = [Z]^T [M] [Z]$ may be ill-conditioned if the columns of $[Z]$ are not sufficiently independent. It does not appear that there is a way to guarantee that the above two conditions will be met with any selection of $[Z]$, however, the chance of them occurring can be minimized with some judicious selection of the vectors. Without proof or discussion, it is to be pointed out here that if the elements of the vector or of matrix $[Z]$ are randomly generated, it has been found that the chances of the above two conditions being violated is very remote.

(ii) Number of Modes Used - An increase in the number of modes used will, in general, decrease the number of iterations required for convergence. However, if more modes are used, the computer time for each iteration will increase because of the increase in sizes of the matrices used; hence, there is a tradeoff.

(iii) Repression of Higher Modes - As pointed out earlier, exact results can be obtained for the first "m" modes of the structure if the trial vectors in $[Z]$ do not contain any contribution from modes higher than "m". Generalizing, it can be said that an improved set of trial vectors can be calculated by suppressing the contribution from the higher modes in the approximate mode shapes at each step. This is achieved as follows.

$$[Z]_j = [K]^{-1} [M] [Z]_{j-1} \quad (59)$$

The subscript j denotes the iteration number. If this iteration is repeated sufficient number of times, modes corresponding to the lowest frequency will be reached. If this iteration is repeated too many times, the mode will repeat itself in one or more columns of $[Z]$ and will render $[Z]^T [M] [Z]$ to be ill-conditioned.

Its use here is not to converge to a mode but just to repress the higher modes and, hence, just a one time application is advisable.

However, in this case $[K]^{-1}$ is required for which $[K]$ has to be non-singular. Thus, the technique can be applied only if $[K]$ is not singular or has been made such with some technique as described next.

(iv) Shifting - Shifting is an useful technique to speed the convergence of modes whose eigenvalues are close to the shift value. As an additional benefit of shifting process is the conversion of the stiffness matrix (in case of a free-free structure) from singular to a non-singular matrix. The method is as follows.

The eigenvalue problem is

$$[K][\Phi] = [M][\Phi][\omega^2] \quad (60)$$

where: different quantities carry their usual meaning.

Also, it has to be noticed that $[K]$ may be singular. To introduce the shift value, λ_s , the following operation is performed. The quantity, $\lambda_s[M][\Phi]$ is subtracted from both sides of equation (60). Thus

$$([K] - \lambda_s[M])[\Phi] = [M][\Phi]([\omega^2] - \lambda_s[I]) \quad (61)$$

By definition

$$[\hat{K}] = [K] - \lambda_s[M] \quad (62)$$

and

$$[\Omega^2] = [\omega^2] - \lambda_s[I] \quad (63)$$

Therefore, final equation is

$$[\hat{K}][\Phi] = [M][\Phi][\Omega^2] \quad (64)$$

This is now the eigen-problem to be solved rather than (60). It is to be noticed that $[\hat{K}]$ is non-singular even if $[K]$ was not.

The eigenvalues of the original system are easily obtained as

$$\omega_i^2 = \Omega_i^2 + \lambda_s \quad (65)$$

The convergence will be to the lowest absolute value of ω^2 . Thus, shifting by a value, λ_s , the eigenvalues, ω^2 , around this shift point are converged to first.

Some general remarks on Shift:

- (a) Analysis of a Free-Structure - Because a free structure has a singular stiffness matrix, the solution of the simultaneous equations in the iteration loop is not possible. However, the shift technique alleviates the problem.
- (b) Specific Frequency Range - When a shift value is used, the modes with eigenvalues closest to the shift value will converge first, which enables one to obtain the modes in the desired frequency range only.
- (c) Large number of modes - By repeated use of different shift values, any number of modes can be obtained.
- (d) The following observations are made without discussion.
 - (i) If the lowest eigenvalues in the range $\omega_1^2, \omega_2^2, \dots, \omega_k^2$, are needed, a shift value of zero should be used for a restrained structure and one for a free-free structure.
 - (ii) If the modes are needed in an intermediate range, a shift midway between the lowest and the highest expected eigenvalues should be used.

2.2.2 Mass and Stiffness Matrices - The iterative Rayleigh-Ritz analysis subroutines require as input mass and stiffness matrices of a structure. To reduce engineering time required to perform an analysis, subroutines were included to calculate mass and stiffness matrices for general standard structural elements. The basic idea behind the subroutines for mass and stiffness is outlined here for continuity.

Mass and stiffness matrices of the complete structure (fluid and bladder) are calculated using finite-element approach. In this approach, a continuous structure (fluid and bladder each separately) is assumed to be composed of simple, small structural elements - the so called finite elements - such as tetrahedron, pentahedron, triangular plates, quadrilateral plates, etc. The procedure to obtain the finite-element mass and stiffness matrices is based on kinetic and strain energy principles, respectively.

The kinetic energy for a complete structure may be expressed as

$$T = \frac{1}{2} \iiint \rho(x, y, z) \cdot \dot{\delta}^2(x, y, z, t) dx dy dz \quad (66)$$

where: T = kinetic energy
 ρ = mass density
 $\dot{\delta}$ = time rate change of deflection
 t = time
 x, y, z = global coordinates

The difficulty in integrating equation (66) is expressing the deflection $\dot{\delta}(x, y, z, t)$ as a continuous function over the complete structure. In the finite-element approach, however, this apparent difficulty is circumvented by idealizing the structure to be comprised of many small structural elements for which $\dot{\delta}(x, y, z, t)$ can be expressed as a continuous function. Thus, the expression (66) is valid for each of the finite-elements of the structure. Then the kinetic energy of the structure is the summation of the kinetic energies of each of the finite elements, that is,

$$T = \sum T_i \quad (67)$$

where i refers to one particular finite element "i".

The common junction of finite elements is denoted as panel points, nodes or joints. Now, however, the deflection $\dot{\delta}(x, y, z, t)$ is easily expressed as a simple function of the joint deflections. These element joint deflections are then generalized coordinates or degrees of freedom of the complete structure.

The approach is to derive the mass matrix for finite-element, "i", in a convenient local coordinate system and then transform it to the Global coordinate system. The technique is outlined here:

$$T_i = \frac{1}{2} \left\{ \dot{\delta}_L(t) \right\}_i^T [m_L]_i \left\{ \dot{\delta}_L(t) \right\}_i \quad (68)$$

where $[m_L]_i$ = the mass matrix in the local coordinate system for the ith element. This mass matrix is obtained by integration using an assumed displacement function. The discussion is deferred till later.

$\{\dot{h}_L(t)\}_i$ = the time rate of change of the joint deflections of finite-element, "i". This is in local system.

The deflections in the local coordinate systems are related to deflections in the global coordinate directions by a transformation matrix, $[T]_i$. Thus

$$\{\dot{h}_L(t)\}_i = [T]_i \{\dot{h}_G(t)\}_i \quad (69)$$

where $\{\dot{h}_G(t)\}_i$ = the joint deflections of finite element, "i", in the global coordinate system.

Using equation (69) in equation (68)

$$T_i = \frac{1}{2} \{\dot{h}_G(t)\}_i^T [m_G]_i \{\dot{h}_G(t)\}_i \quad (70)$$

$$\text{where } [m_G]_i = [T]_i^T [m_L]_i [T]_i \quad (71)$$

is the mass matrix with respect to the global coordinate system for the ith finite-element. Further, all the elemental mass matrices are finally assembled to give the mass matrix of the total structure, as shown in equation (67).

The development of the finite-element stiffness matrices is similar to that of the mass matrices. The strain energy for the structure may be expressed as the summation of the strain energies of each finite elements. That is,

$$U = \sum U_i \quad (72)$$

As was done for the finite-element mass matrix, the stiffness matrix for finite-element, "i", is derived in a convenient local coordinate system. Thus,

$$U_i = \frac{1}{2} \{\dot{h}_L(t)\}_i^T [k_L]_i \{\dot{h}_L(t)\}_i \quad (73)$$

where $[k_L]_i$ = the stiffness matrix with respect to local coordinate directions for finite element, "i". This stiffness matrix is obtained by integration using an assumed displacement function. This will be discussed later.

$\{h_G(t)\}_i$ = the joint deflections of finite element, i, measured in local coordinate system.

The same transformation matrix, $[\gamma]$, which was used in equation (70) is used here to relate the deflections in local coordinates to deflections in global coordinates. Substitute then,

$$U_i = \frac{1}{2} \{h_G(t)\}_i^T [k_G]_i \{h_G(t)\}_i \quad (74)$$

where

$$[k_G]_i = [\gamma]_i^T [k_L]_i [\gamma]_i \quad (75)$$

is the stiffness matrix with respect to the global coordinate system for the ith element.

Euler angle rotations at some joints (where the body coordinate is needed to be different than that of the global coordinates) are input in the program to allow the joint degree of freedom at these points to be different than that of global x,y,z directions.

However, in case of fluid, there is another item to be taken care of as far as stiffness matrix is concerned. This has to do with the surface elements of the fluid. The item concerned is known as gravitational potential. The energy contributed is known as the gravitational potential. This is caused by fluid movement in the gravitational field. Development of the gravitational potential stiffness effect is given in Section 2.2.4.

2.2.3 Stiffness and Mass Matrices in Local Coordinates (Solid Element)

Mass Matrix (Triangular Element)

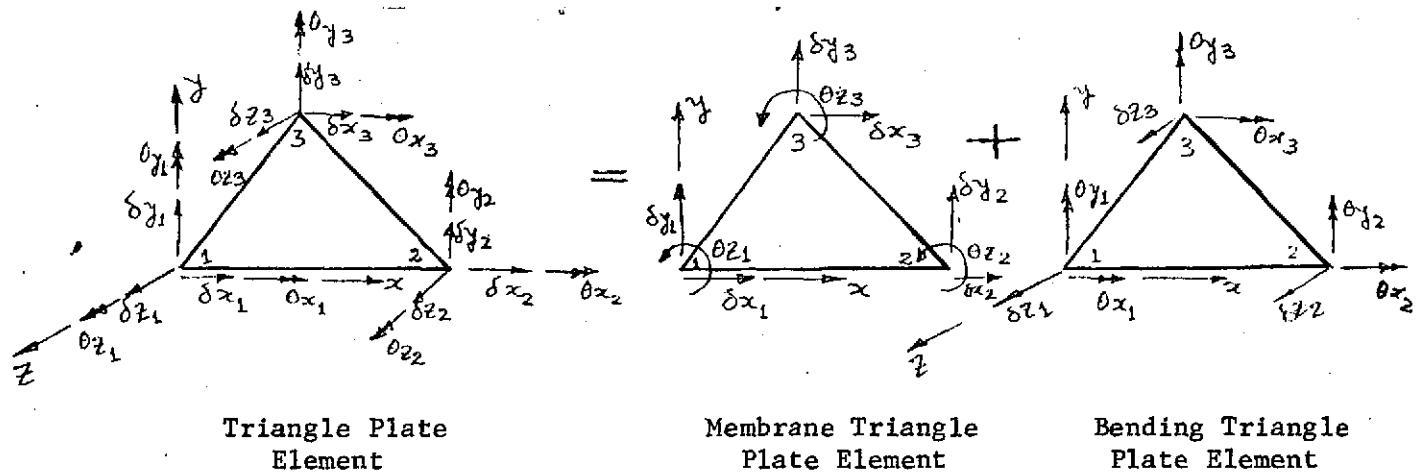


Figure - 2

The consistent mass matrix for a triangle plate element are calculated as a combination of the two following elements.

- (i) consistent mass matrix for a membrane triangle plate element,
- (ii) consistent mass matrix for a bending triangle plate element.

The elements of the mass matrix for the membrane triangle plate element represent the distributed mass properties of the triangle. These matrix elements are calculated by assuming a quadratic displacement field⁽³⁾. The elements of the mass matrix for the bending triangle plate element represent the distributed mass properties of the triangle. These matrix elements are calculated by assuming a cubic displacement field⁽⁴⁾.

Stiffness Matrix: (triangular element)

The stiffness matrix for the triangle plate element is calculated in the same manner as the mass matrix and the technique and displacement fields are exactly the same.

Mass Matrix (quadrilateral element)

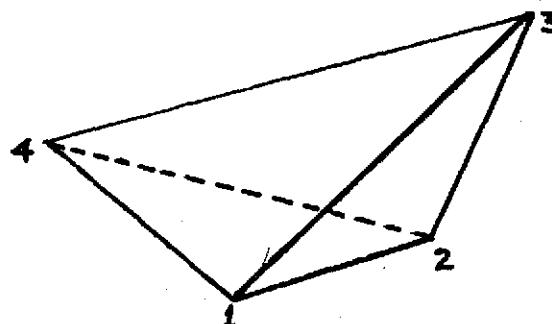


Figure 3

This mass matrix is calculated by taking the average of the four overlapping triangles created by the diagonals (1.3) and (2.4). The triangles are handled as discussed before. Thus, the quadrilateral case is nothing but a combination of triangular case.

Stiffness Matrix (quadrilateral element)

This is handled in exactly same fashion as the mass matrix case.

2.2.4 Stiffness and Mass Matrices in Local Coordinates (fluid finite element)

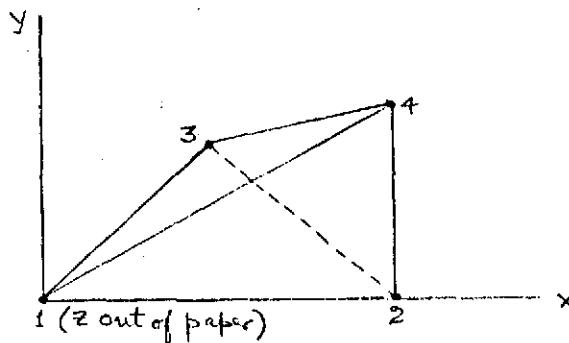


Figure - 4
Local Coordinate System
for Tetrahedron Element

The basic fluid element is a tetrahedron; pentahedron elements and hexahedron elements are synthesized, simply by placing six and ten overlapping tetrahedrons together respectively and averaging the result. The averaging is carried out to eliminate the bias, if any.

For each tetrahedron element, a local cartesian coordinate system is defined so that vertex 1 is the origin, the x-axis includes vertex 2, vertex 3 lies in the x-y plane and vertex 4 always has a positive Z-coordinate (figure above).

This element considers a linear displacement field (constant strain). This is boundary conformable. The displacement field throughout the element is expressed in terms of coordinate locations and appears as

$$\bar{\omega}(x, y, z, t) = \bar{a}_0 + \bar{a}_1 x + \bar{a}_2 y + \bar{a}_3 z \quad (76)$$

The coefficients $\bar{a}_k(t)$, $K = 0, 1, 2, 3$ are eliminated in terms of the 12 vertex displacements.

The mass matrix for the fluid elements is obtained by expressing the kinetic energy

$$T = \frac{1}{2} \int_{Vol} \dot{\bar{\omega}} \cdot \dot{\bar{\omega}} \rho dV \quad (77)$$

where: T = kinetic energy

$$\dot{\bar{\omega}} = \dot{\bar{a}}_0 + \dot{\bar{a}}_1 x + \dot{\bar{a}}_2 y + \dot{\bar{a}}_3 z$$

ρ = mass density

This gives rise to a (12x12) mass matrix.

The stiffness matrix for the fluid element is obtained by expressing volumetric dilatation strain energy and gravitational potential energy in terms of vertex displacement coordinates

$$U_D = \frac{1}{2} \int_{Vol} k \theta^2 dV \quad (78)$$

$$U_g = \frac{1}{2} \rho g \int_{Area} (\bar{\omega} \cdot \bar{n}) (\bar{\omega} \cdot \bar{e}) d\sigma \quad (79)$$

where U_D = volumetric dilatation energy

U_g = gravitational potential energy

k = fluid bulk modulus

θ = volumetric strain

\bar{n} = unit outer normal

\bar{e} = a unit vector parallel with the gravity vector \bar{g} , but of opposite sense, i.e., $\bar{e} = -\bar{g}/g$

An observation is made with respect to the gravitational potential energy as expressed in (79). Since it is a boundary conformable element, the surface integrals such as (79) will all cancel each other throughout the interior of the fluid in a container, since \bar{n} on common element boundaries is equal and opposite. Thus, the gravitational potential energy will depend only on displacement coordinates at the boundary of the entire volume of the fluid, the free surface, the wetted container wall and also, in this case, the bladder. Also, for a rigid tank, $\bar{\omega} \cdot \bar{n}$ is non-zero only at the free surface where $\bar{e} = \bar{n}$; thus,

$$U_g = \frac{1}{2} \rho g \int_{\text{Free Surface}} (\bar{\omega} \cdot \bar{n}) ds \quad (80)$$

Stiffness coefficients corresponding to gravitational potential and volumetric strain energies are thus derived.

2.3 Mechanical Equivalent - The mechanical equivalent in this case has been calculated on the following basis. The forces and moments developed by the model due to an external disturbance should correspond to the forces and moments exerted by the fluid under similar conditions.

Since any arbitrary liquid motion can be thought of as superposition of different slosh modes it suggests itself that the mechanical equivalent must consist of a series of spring mass systems, the masses corresponding to the effective amounts of liquid oscillating in different slosh modes. The frequencies of the mechanical equivalent must correspond to the frequencies of the elastomer for the mode they represent. However, to account for the part of the fluid that does not take part in the motion, one may also add a mass without a spring.

A short definition of the mechanical equivalent is as follows:

The equations of motion for a base driven elastic system are

$$\{\ddot{\xi}\} + [2\zeta\omega]\{\dot{\xi}\} + [\omega^2]\{\xi\} = -[\Phi]^T [M][\tau]\{\ddot{q}_b\} \quad (81)$$

where:

ξ : modal coordinates,

$\dot{\xi}$: modal velocities,

$\ddot{\xi}$: modal accelerations,

Φ : discrete mode shapes,

τ : rigid body transformation matrix,

\ddot{q}_b : discrete base accelerations (6 degrees of freedom),

ζ : modal damping,

ω : frequencies,

M : mass matrix

The base reactions to support a given modal accelerations are

$$\{F\}_{\text{base}} = [T]^T [M] \{\phi\} \ddot{\xi} \quad (82)$$

Noting that

$$\ddot{\xi} \propto \{\phi\}^T [M] [T] \{\ddot{q}_6\} \quad (83)$$

Substituting from (83) in (82)

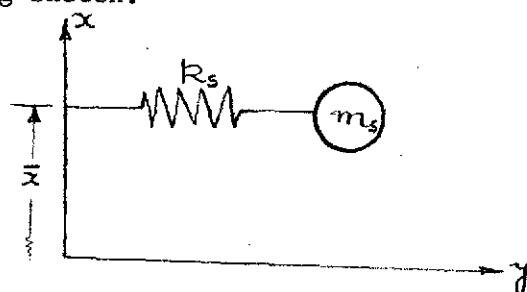
$$\{F\}_{\text{base}} \propto [T]^T [M] \{\phi\} \{\phi\}^T [M] [T] \{\ddot{q}_6\} \quad (84)$$

where the proportionality parameter is a function of frequency.

The product

$$[T]^T [M] \{\phi\} \{\phi\}^T [M] [T]$$

is a (6x6) matrix, in general, for a given mode. This may be called a matrix of forces and moments. The equivalent slosh parameters can be interpreted as shown in the following sketch.



where
 k_s : effective stiffness
 m_s : effective slosh mass
 \bar{x} : distance of line of action from the reference point.

The reference point is the same as the reference for the rigid body transformation.

This equivalent model is only valid for loads at the reference point. This may not be used for getting any detailed information concerning what is happening inside the elastic system.

In this particular case, the final product is a matrix (3x3). The term (2,2) is the slosh mass in the y-direction and the ratio of terms (2,3) and (2,2) is the distance \bar{x} . It should be noted that \bar{x} may be outside the tank boundary. The slosh mass and \bar{x} are printed out in the computer run. The slosh stiffness is the frequency squared because the modal displacements are normalized such that the generalized mass is unity. To obtain the mechanical equivalent slosh parameters of the preceding sketch, only y-translation and θ_z rotation need be considered in the rigid body transformation matrix T . Thus, the general 6x6 slosh matrix is a 2x2 matrix for the particular case considered here. The 1,1 term is the slosh mass in the lateral y-direction and the ratio of terms (1,2) and (1,1) is the distance \bar{x} . It should be noted that \bar{x} may be outside the tank boundary. The slosh stiffness is the slosh frequency squared times the slosh mass, $\omega_s^2 m_s$.

3. MODEL

The modeling of the system is achieved in two distinct parts. First, the static free surface of the fluid/bladder system, within the constraints, is established (refer to 2.1). This static free surface is subsequently used in modeling the total system as an assemblage of finite-elements for vibration.

3.1 Finite Element Model of the System - This is an axisymmetric system. The system thus can be subdivided into four quadrants, 1, 2, 3 and 4, as shown in the figure. Quadrants 2, 3 and 4 are reflected on to quadrant 1 with the following technique whereby the analysis of only one quadrant is necessary for the dynamic analysis of the total system. This renders the problem relatively small as far as sizes of the matrices are concerned.

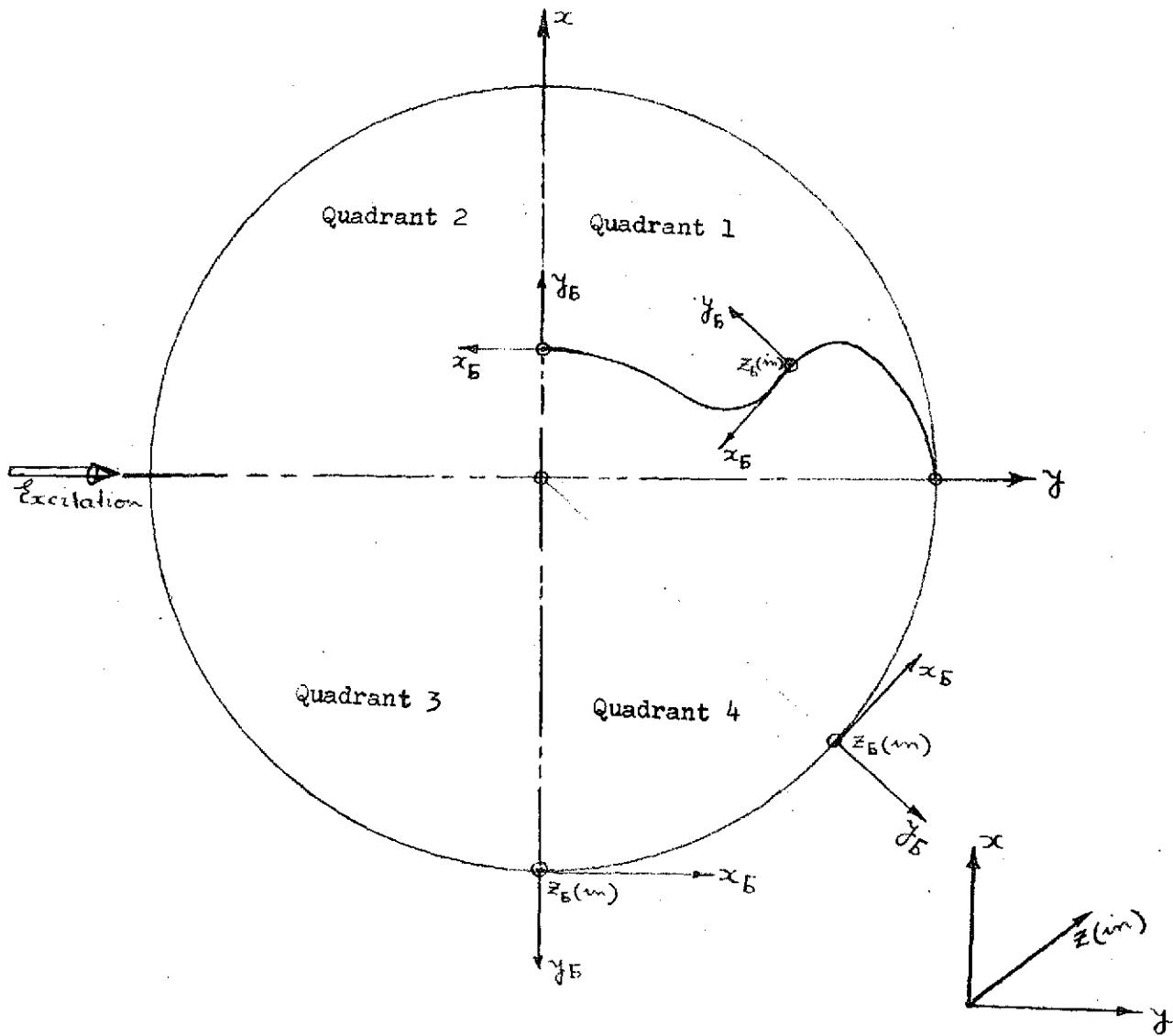


Figure 5. Quadrants and Coordinates Systems

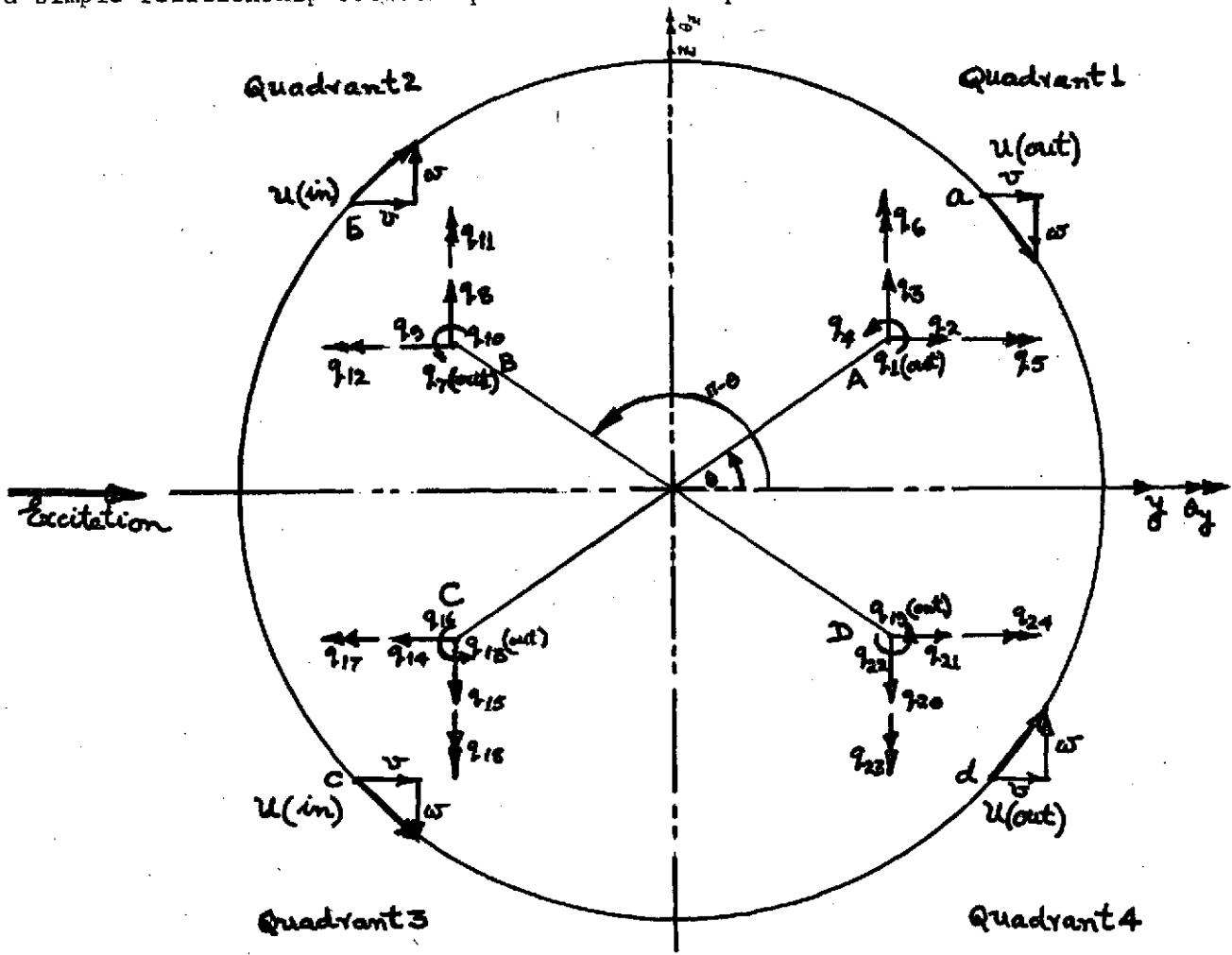
$x \ y \ z$ is the global coordinate system (right handed)

$x_b \ y_b \ z_b$ is the so called body coordinate system (right handed)

The global and body coordinate system systems are two different coordinate systems related through the Euler rotations. The global system remains fixed but the body coordinate system varies for different points.

The global system is used for interior fluid and the body coordinate system is used on the boundary. The use of body coordinate system on the boundaries makes the handling of the boundary condition easier. The body coordinates are shown in the figure for a few important points. The reflections of quadrants and the derivation of the Euler angles are discussed in the following pages.

3.2 Reflections of Quadrants 2, 3 and 4 on Quadrant 1 for Lateral Slosh - For the case of lateral slosh and the excitation direction a simple relationship between q 's of different quadrants are as follows.



Lateral Slosh (Interior Points)

Figure 6

where: u, v and w are the velocities in the global x, y and z directions.

and a : a typical point in quadrant 1 (on the boundary)

b : a typical point in quadrant 2 (on the boundary)

c : a typical point in quadrant 3 (on the boundary)

d : a typical point in quadrant 4 (on the boundary)

q 's: local coordinates

A: a typical point in quadrant 1 (interior)

B: a typical point in quadrant 2 (interior)

C: a typical point in quadrant 3 (interior)

D: a typical point in quadrant 4 (interior)

$$\begin{Bmatrix} q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{11} \\ q_{12} \end{Bmatrix}_2 = \begin{Bmatrix} -1 & & & & & \\ & -1 & & & & \\ & & -1 & & & \\ & & & 1 & & \\ & & & & -1 & \\ & & & & & 1 \end{Bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix}_1 \quad (85)$$

$$\begin{Bmatrix} q_{13} \\ q_{14} \\ q_{15} \\ q_{16} \\ q_{17} \\ q_{18} \end{Bmatrix}_3 = \begin{Bmatrix} -1 & & & & & \\ & -1 & & & & \\ & & -1 & & & \\ & & & -1 & & \\ & & & & -1 & \\ & & & & & -1 \end{Bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix}_1 \quad (86)$$

$$\begin{Bmatrix} q_{19} \\ q_{20} \\ q_{21} \\ q_{22} \\ q_{23} \\ q_{24} \end{Bmatrix}_4 = \begin{Bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & -1 & & \\ & & & & -1 & \\ & & & & & -1 \end{Bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix}_1 \quad (87)$$

Thus, it is apparent

$$\{q\}_3 = - \{q\}_1 \quad (88)$$

$$\{q\}_4 = - \{q\}_2 \quad (89)$$

There exists a relation between q's and global coordinates. They are (writing them in terms of their velocity or displacement components)

$$\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix}_1 = \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} \quad (90)$$

$$\begin{Bmatrix} q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{11} \\ q_{12} \end{Bmatrix}_2 = \begin{Bmatrix} u \\ w \\ -v \\ \theta_x \\ \theta_z \\ -\theta_y \end{Bmatrix} \quad (91)$$

Recognizing the relations (88) and (89), the other quadrants relations are not followed through any further. From relations (85) and (91) the relation between the degrees of freedom of quadrants 1 and 2 for interior of the fluid can be written as

$$\begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_2 = \begin{bmatrix} -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_1 \quad (92)$$

Thus

$$\{u\}_2 = [T_{21}] \{u\}_1 \quad (93)$$

where

$$\{u\} = \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}$$

$$[T_{21}] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix}$$

If the boundary degrees of freedom were also in the global comparison system, the same transformation could be applied to them. But as the situation stands, they are in the body coordinates system which in general will have a different transformation than $[T_{21}]$.

4. RESULTS

The results of the study has been obtained in three distinct steps. They are: (1) Static Equilibrium Shape, (2) Vibration Analysis and, (3) Equivalent Slosh Mass for Dynamic Characteristics of the System. The static equilibrium shapes obtained in the first part are fed into the second part for establishing the grid pattern for finite-element analysis of the system.

4.1 Static Equilibrium Shapes - The technique used for this has been discussed in Section 2.1. A computer program has been developed for this purpose which is listed in Section 7.

The static equilibrium shapes for fill conditions of 80%, 60%, 50%, 40% and 20% for 1g case have been obtained using the available computer program. However, for the case of $10^{-5}g$ conditions, only fill ratios of 80% and 60% have been successfully run through the program. The other cases of 50%, 40% and 20% were extrapolated from available results. This was achieved with the help of graphical representation between the percentage fill and the coefficients of the power series, assumed for the static surface, see Figures 11G through 14G. The graphical representations of the static free surface and the coefficients are presented in Figures 1G through 14G.

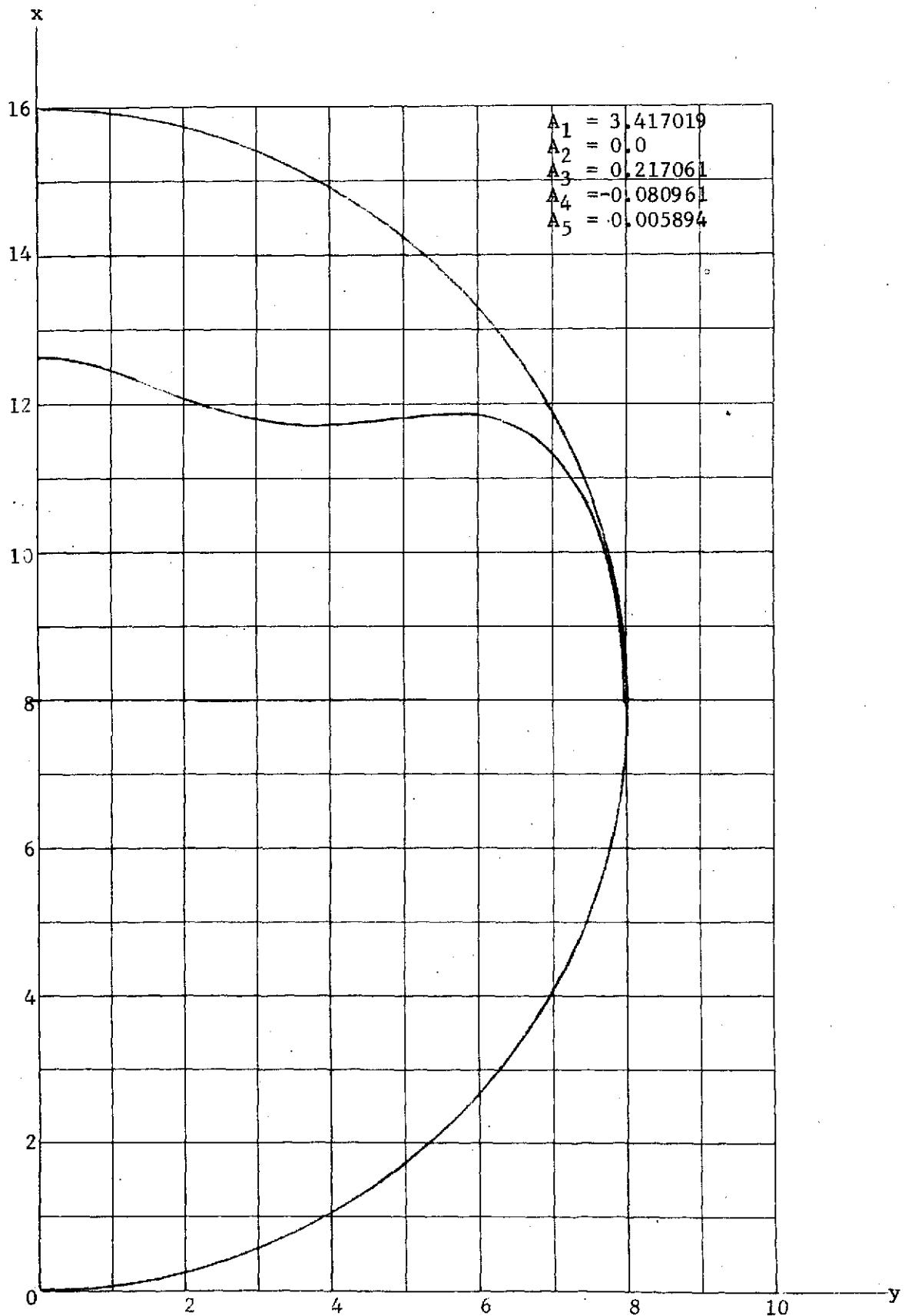
The values of other parameters used in all of these are:

- (i) Radius of the sphere = 8 inches (20.32 cm)
- (ii) Young's modulus for the bladder material = 200 lb/in^2
 $(1.378951 \times 10^6 \text{ N/m}^2)$
- (iii) Thickness of the bladder = 0.06 inches (0.1524 cm)
- (iv) Mass density for mercury = $0.0013 \text{ lb/in/sec}^2$ (0.00059 Kgm)
- (v) Ullage pressure = 0.10 lb/in^2 (689.48 N/m^2)

and the assumptions made are:

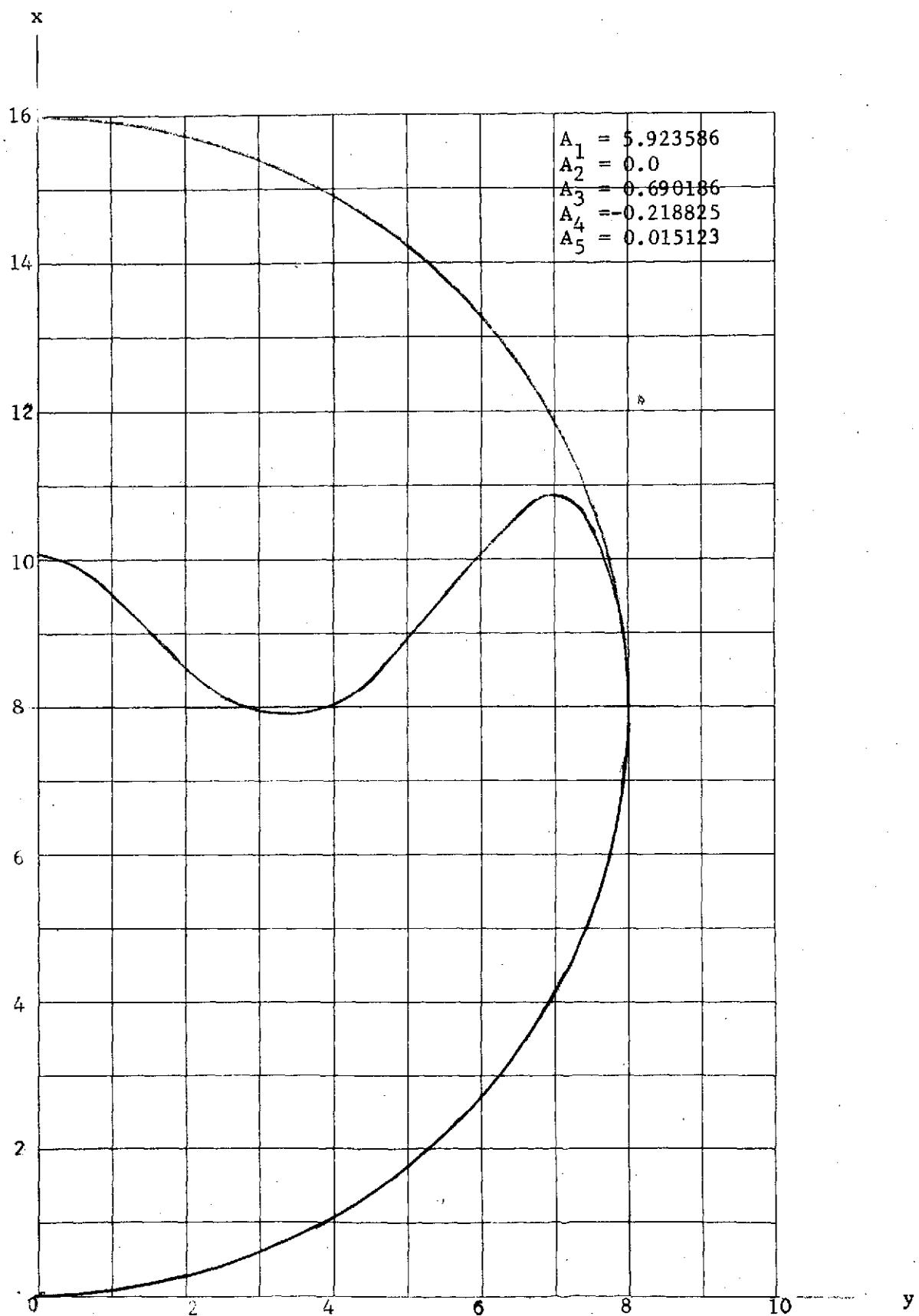
- (vi) Bladder attachment level = diametral plane
- (vii) Order of the polynomial = 4

Figures 11G through 14G may be used to establish the coefficients which define the static free surface for any gravity and any percentage fill without making any more computer runs.



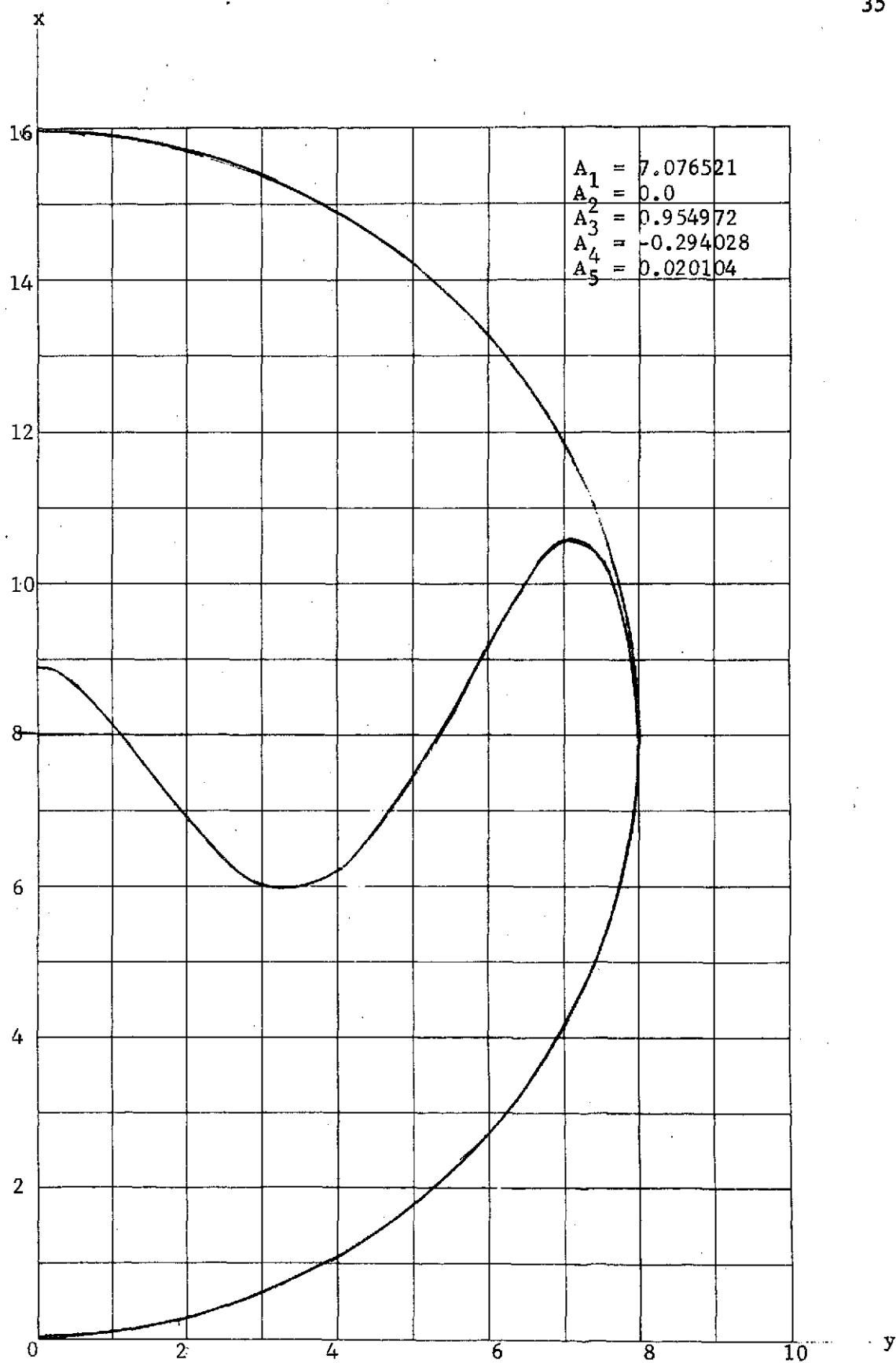
STATIC FREE SURFACE FOR 80% FULL, 1g

FIGURE 1G



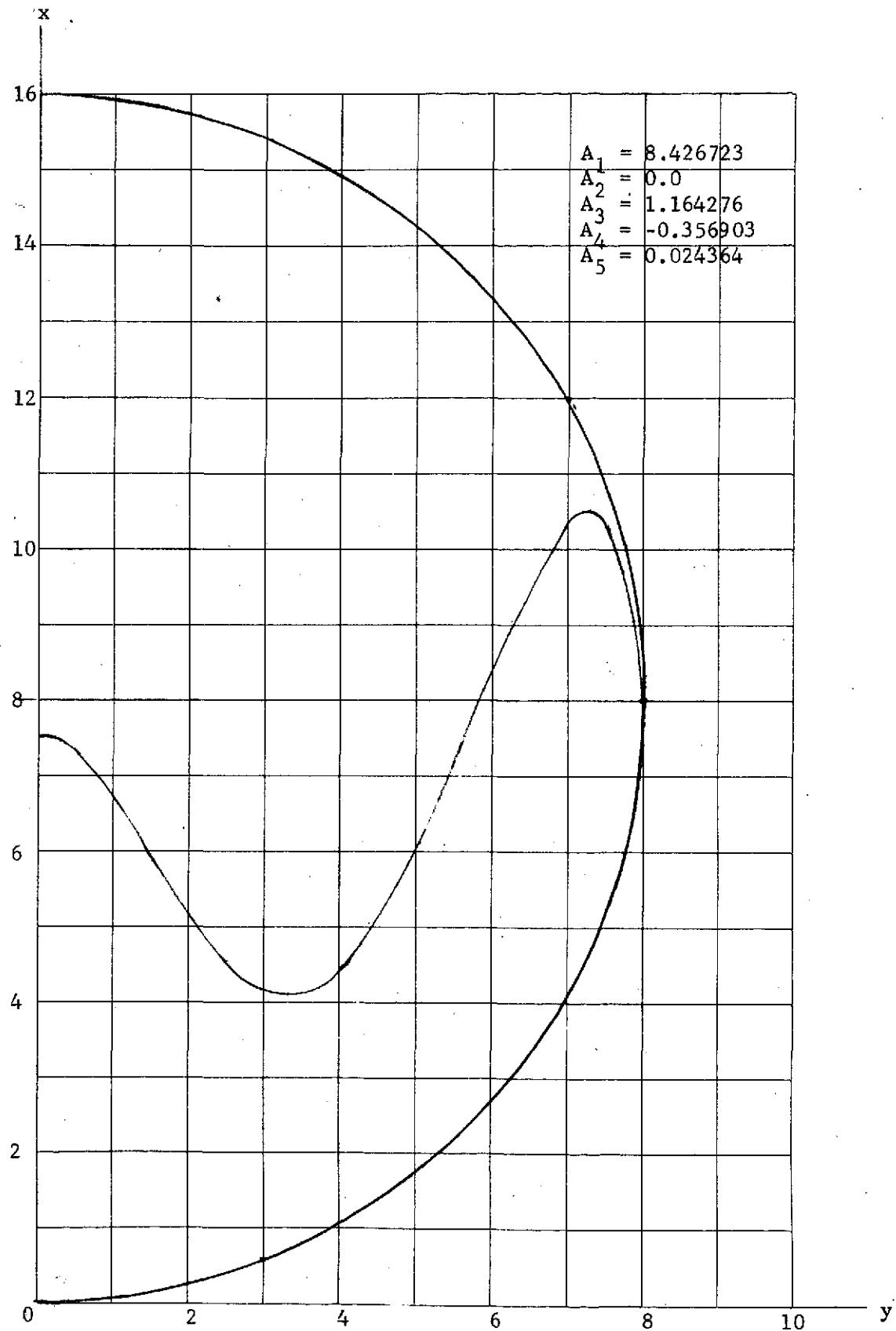
STATIC FREE SURFACE FOR 60% FULL, 1g

FIGURE 2G



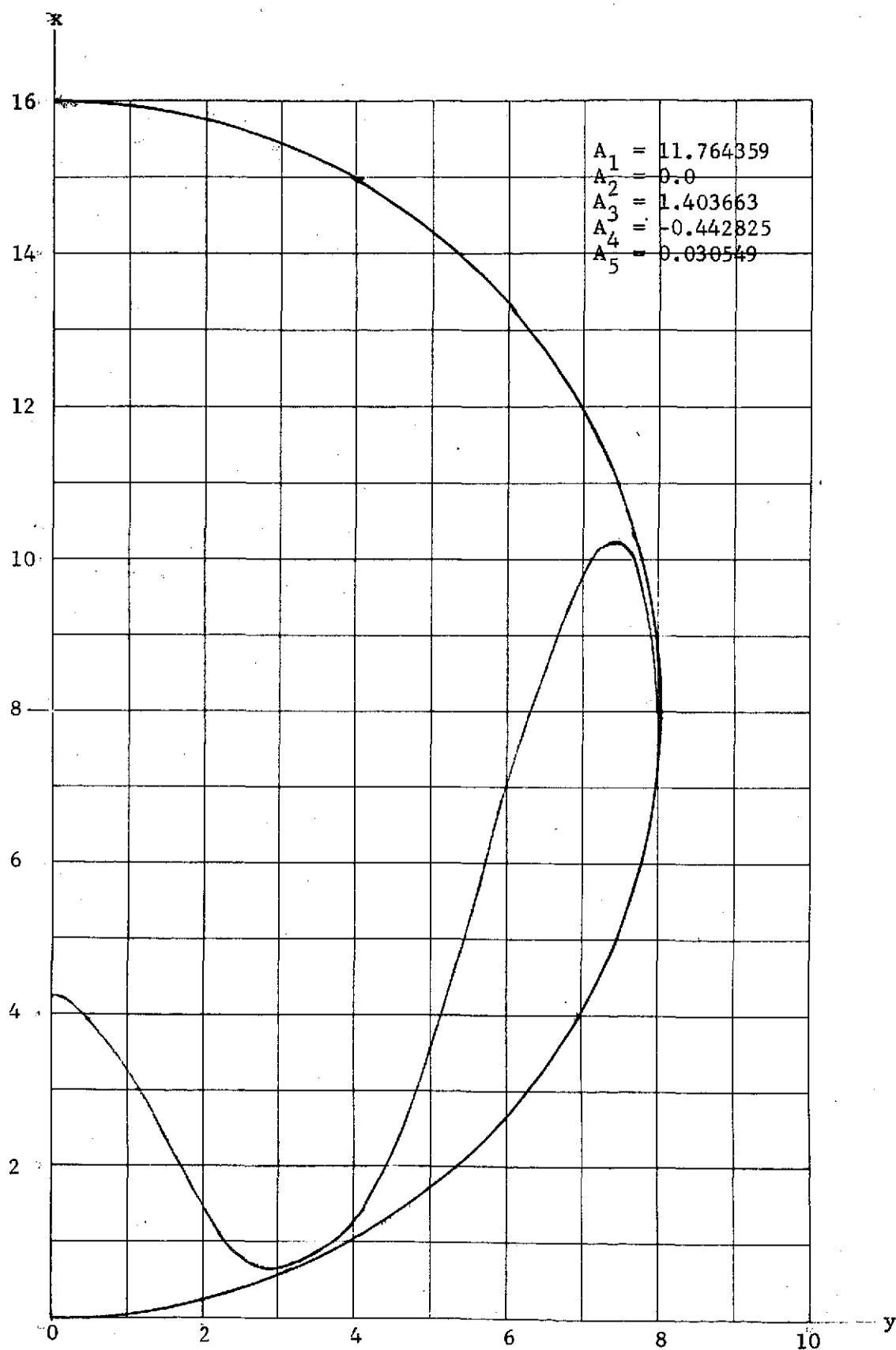
STATIC FREE SURFACE FOR 50% FULL, 1g

FIGURE 3G



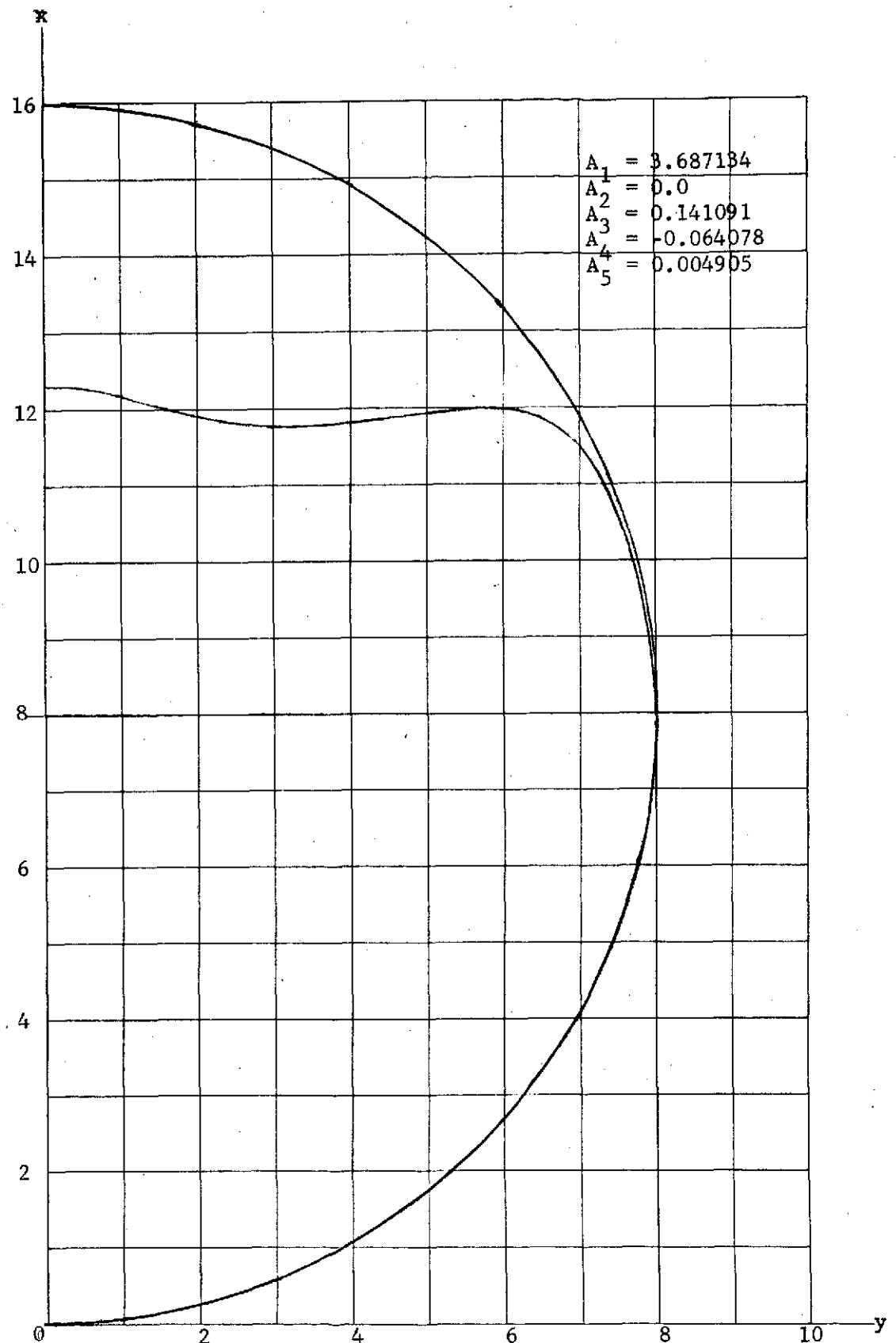
STATIC FREE SURFACE FOR 40% FULL, 1g

FIGURE 4G



STATIC FREE SURFACE FOR 20% FULL, 1g

FIGURE 5G



STATIC FREE SURFACE FOR 80% FULL, $10^{-5} g$

FIGURE 6G

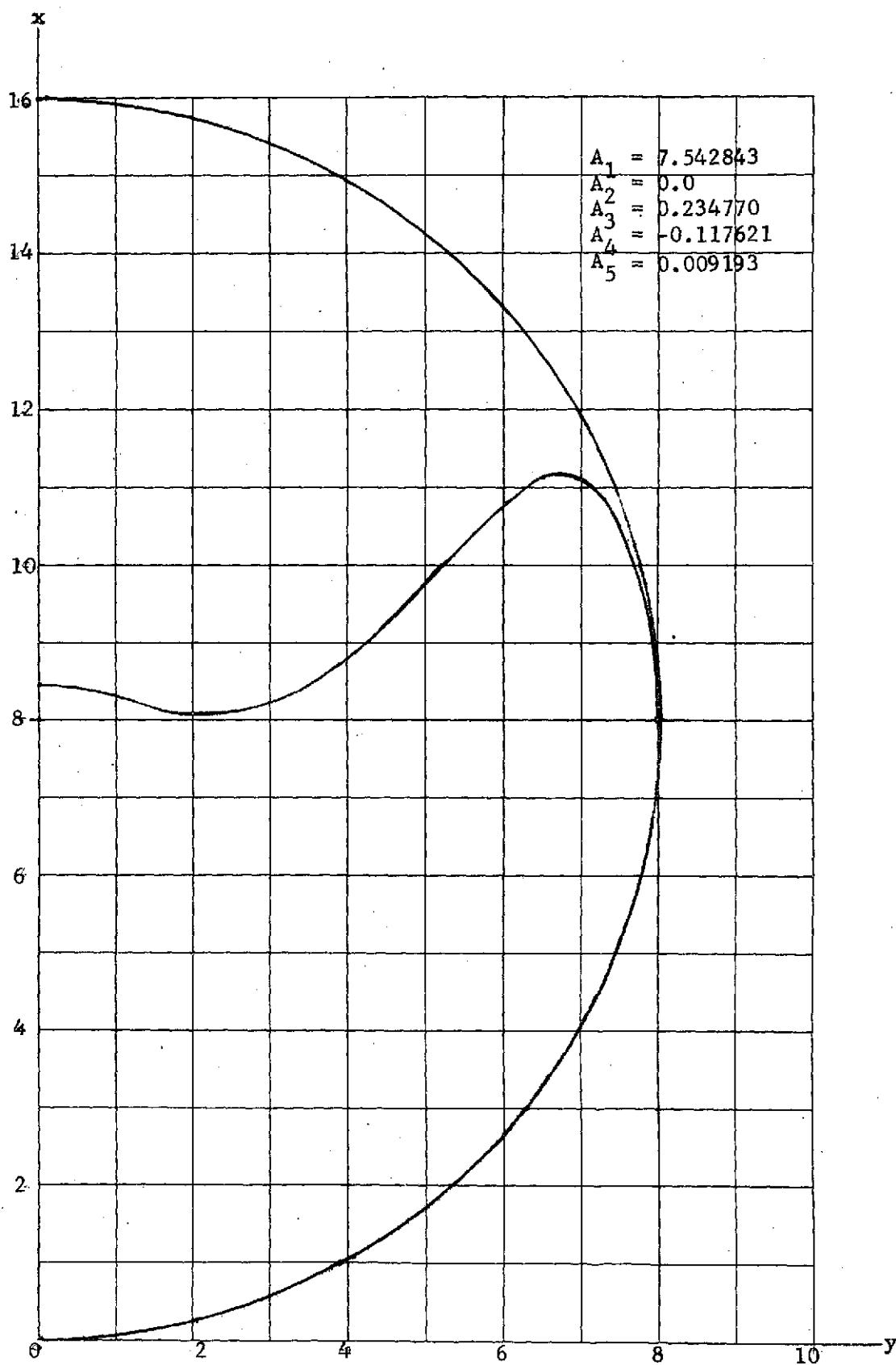
STATIC FREE SURFACE FOR 60% FULL, $10^{-5} g$

FIGURE 7G

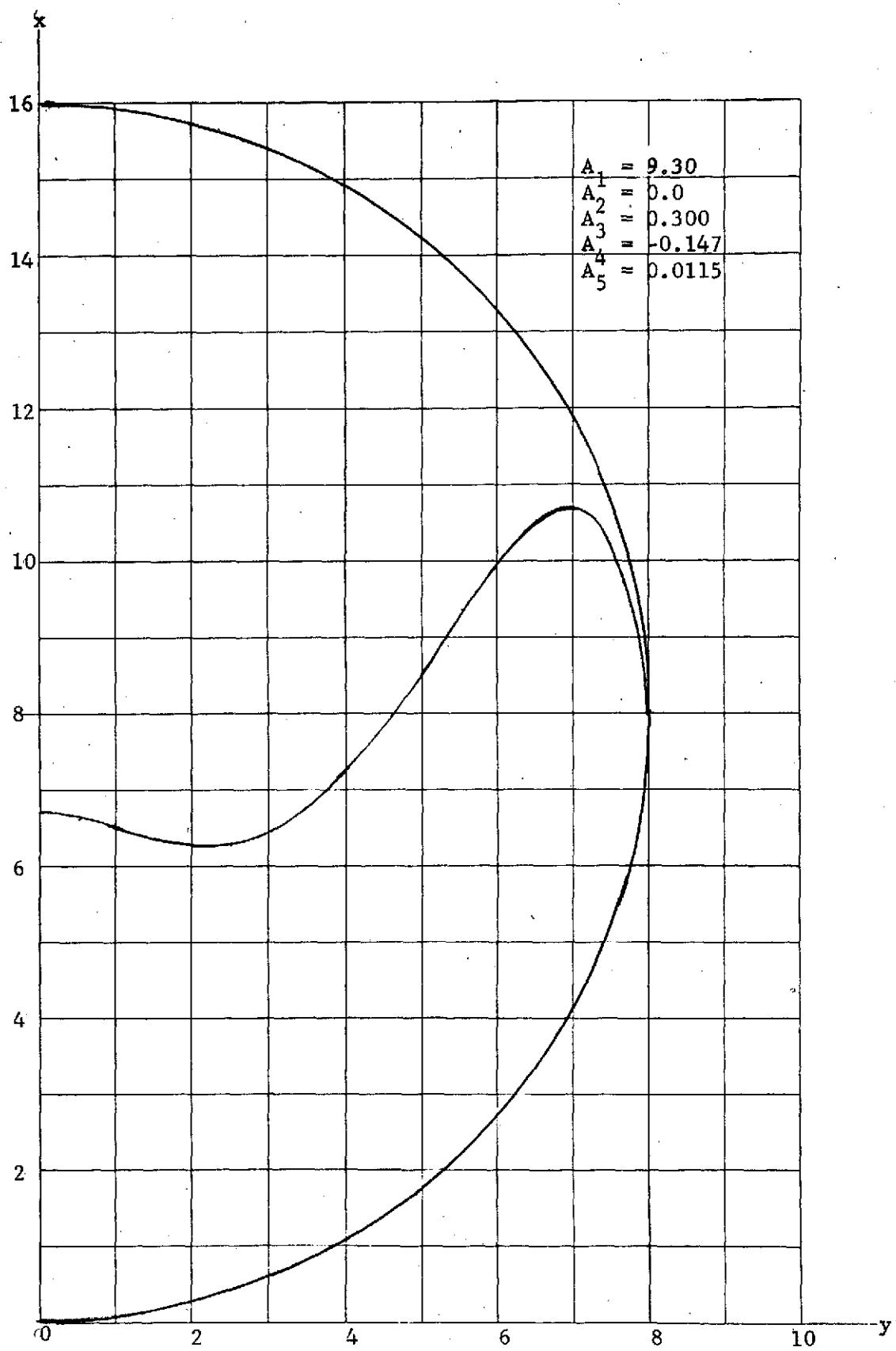
STATIC FREE SURFACE FOR 50% FULL, $10^{-5} g$

FIGURE 8G.

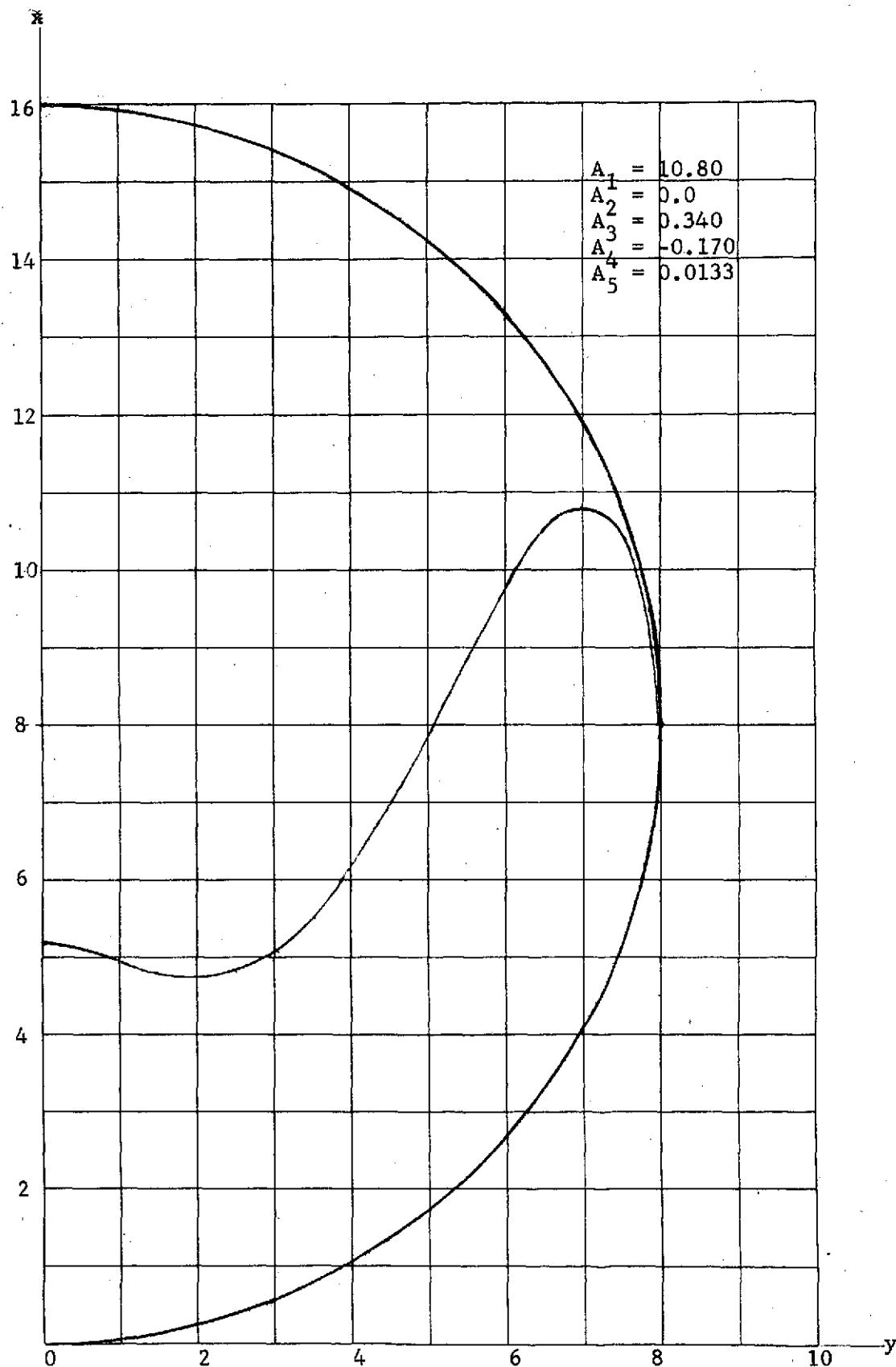
STATIC FREE SURFACE FOR 40% FULL, $10^{-5} g$

FIGURE 9G

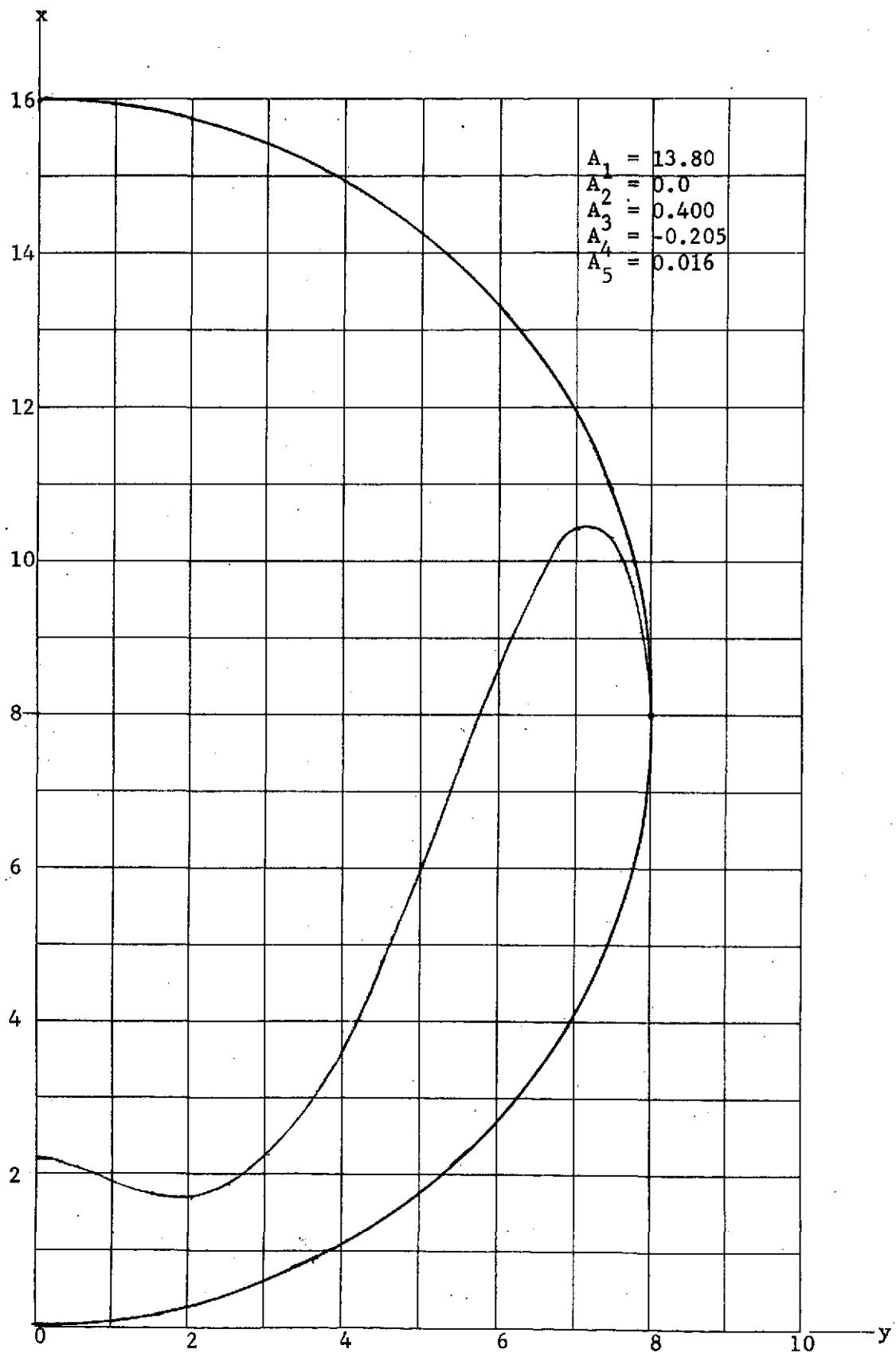
STATIC FREE SURFACE FOR 20% FULL, $10^{-5}g$

FIGURE 10G

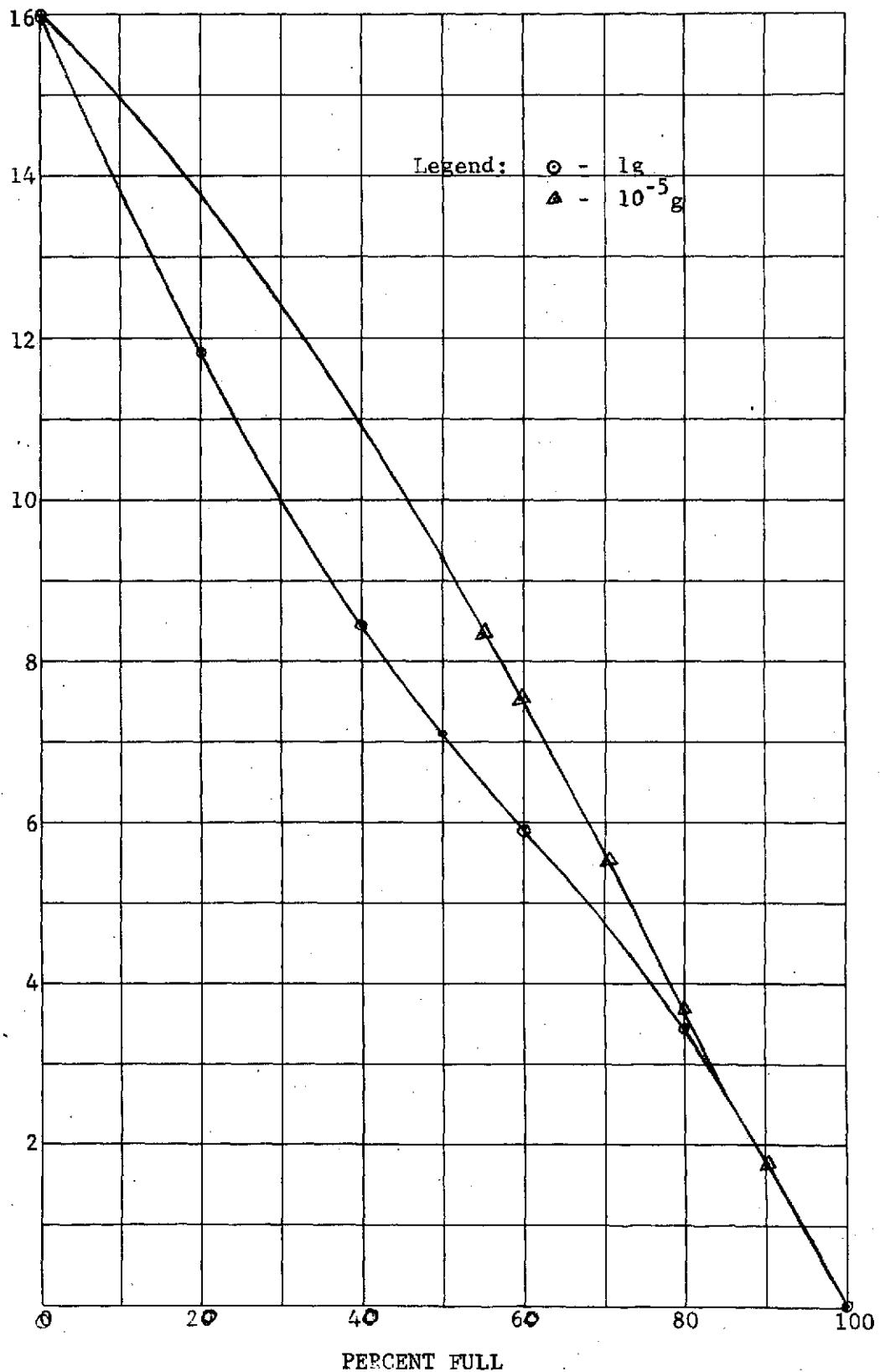
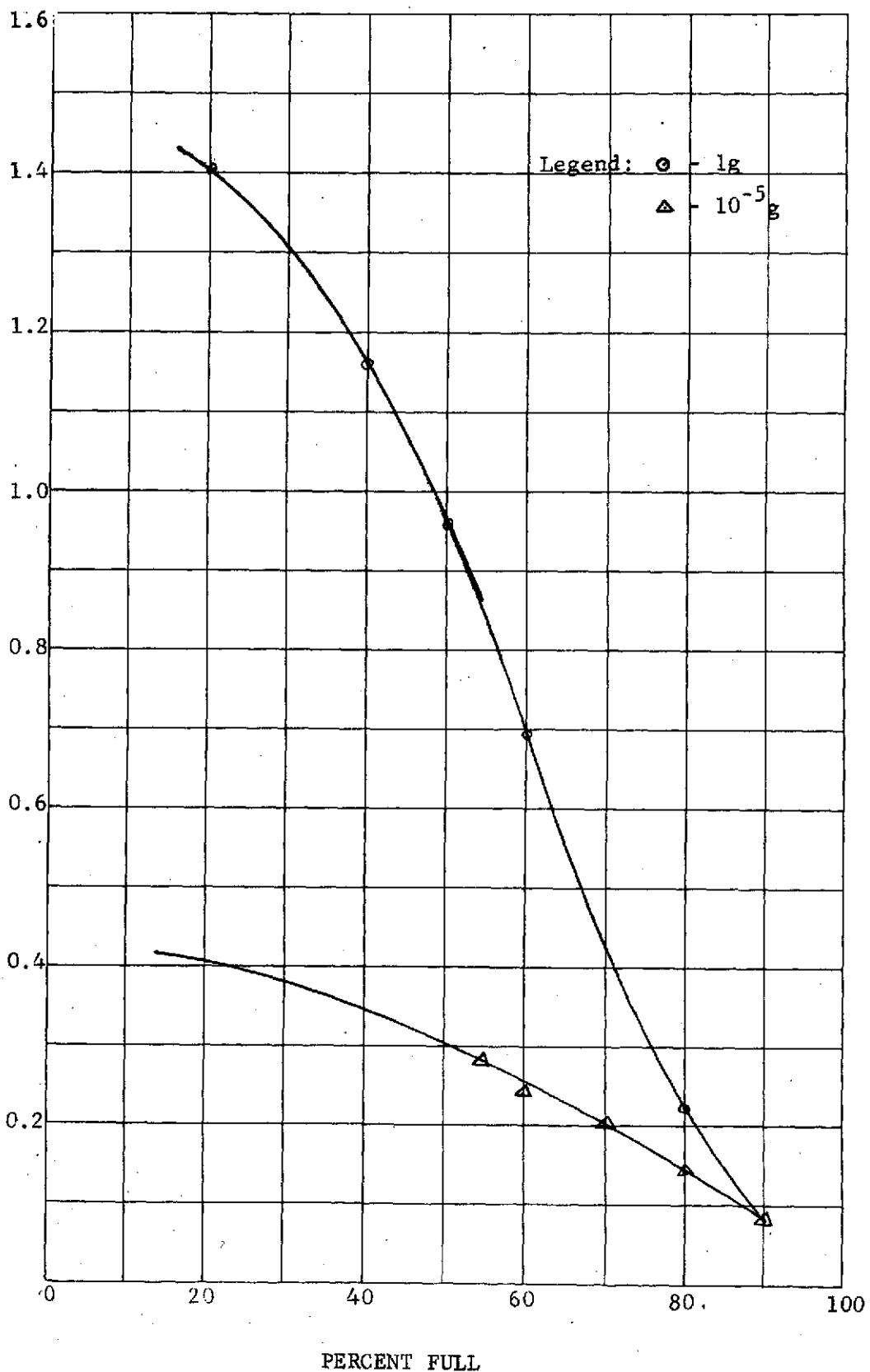
COEFFICIENT A₁ OF THE FOURTH ORDER POLYNOMIAL

FIGURE 11G

COEFFICIENT A_3 OF THE FOURTH ORDER POLYNOMIAL

PERCENT FULL

FIGURE 12G

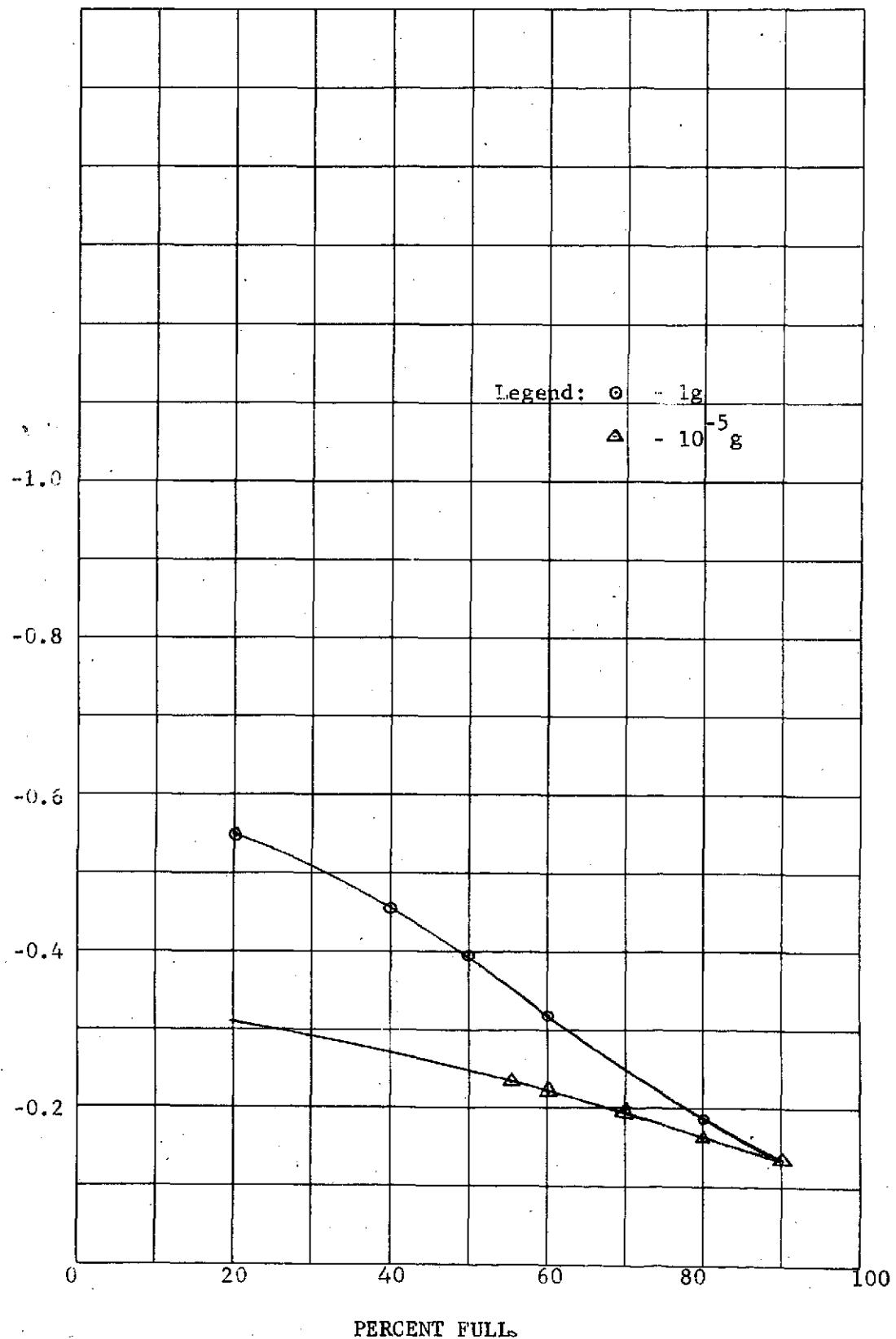
COEFFICIENT A_4 OF THE FOURTH ORDER POLYNOMIAL

FIGURE 13G

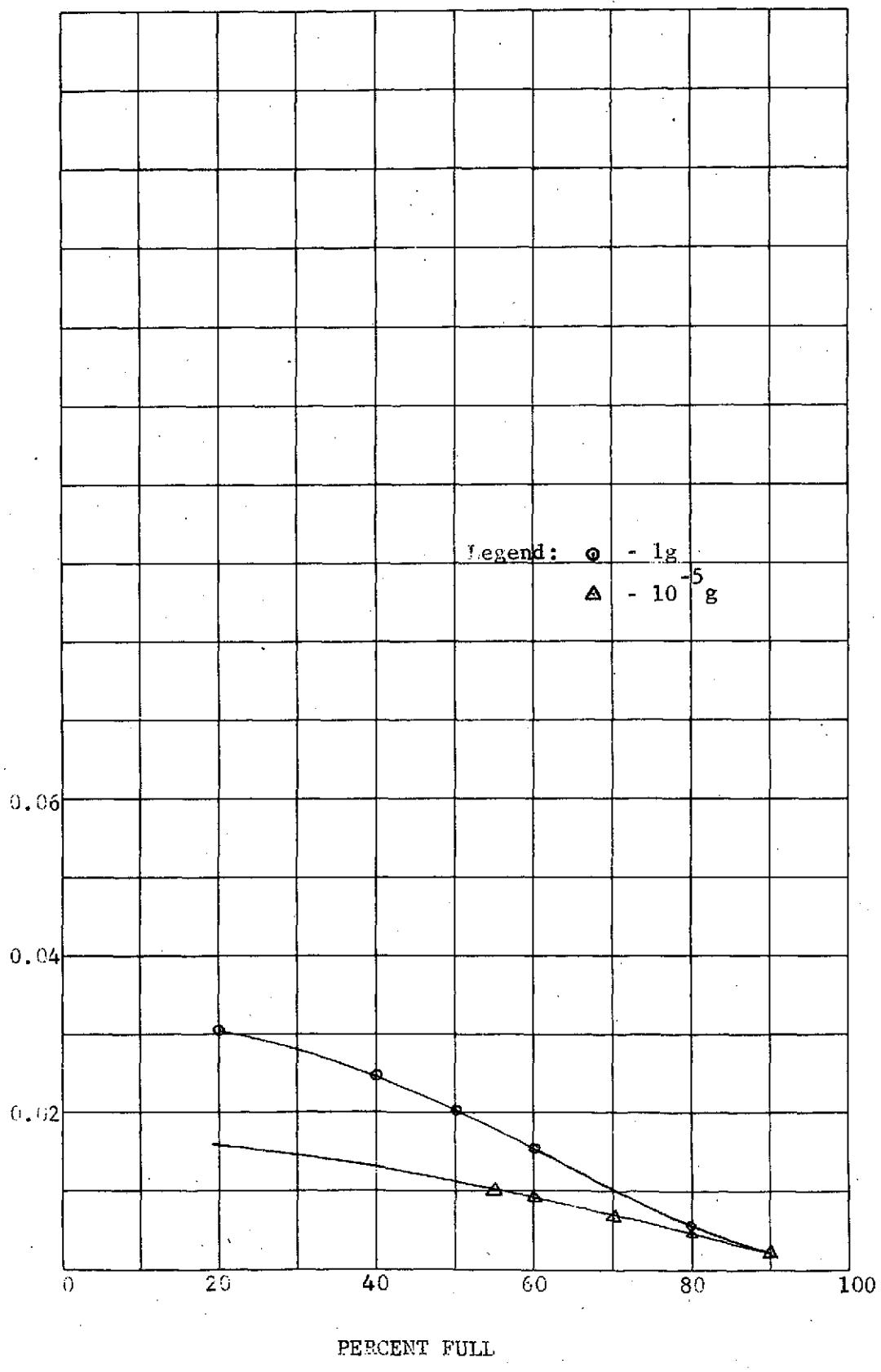
COEFFICIENT A₅ OF THE FOURTH ORDER POLYNOMIAL

FIGURE 14G

4.2 Vibration Analysis - The iterative Rayleigh-Ritz technique used to calculate the vibration mode shapes and frequencies has been described in Section 2.2.1. The computer subroutine "YMODE2" used to calculate these modal values already exists at the MSFC computer center and is, thus, omitted from the program listings of Appendix A2.

The calculated vibration mode shapes were used in the calculation of the mechanical equivalent slosh parameters which are given in Section 4.3. The calculated vibration frequencies are listed in Section 4.3.

4.3 Mechanical Equivalent Slosh - The technique used to calculate the mechanical equivalent slosh parameters has been described in Section 2.3. The computer subroutine "MECHEQ" used to calculate these parameters is included in Appendix A2.

A great deal of difficulty was encountered in the identification of the slosh modes among the many vibration modes calculated. It was decided to identify as slosh modes those modes having the largest slosh mass. The following tables give the slosh frequencies, masses, stiffnesses and attach stations. It should be noted that numerical precision of the computer casts some doubt on the accuracy of these modal results.

Table 4.3-1

SLOSH PARAMETERS
80% FILL CONDITION
1g ACCELERATION

<u>f</u> (Hz)	<u>Mass</u> <u>lb-sec²/in</u>	<u>Stiffness</u> <u>lb/in</u>	<u>Attach Station (x)</u> <u>in</u>
1.0756	.0037	.1690	4.2
1.0828	.0022	.1018	10.0
1.0946	.0002	.0104	6.8
1.1193	.0001	.0044	8.9
1.1235	.0014	.0698	5.0
1.1337	.0011	.0558	6.2
1.1709	.0008	.0411	1.5
1.2429	.0180	1.0978	6.0
1.2471	.0014	.0860	4.3
1.2640	.0250	1.5768	3.9
1.3188	.0130	.8926	5.7
1.3276	.0035	.2435	4.8
1.3709	.0120	.8903	5.1
1.4495	.0007	.0581	11.4
1.4625	.0003	.0262	8.4
1.4729	.0350	2.9976	3.7
1.4952	.0001	.0097	-4.0

Table 4.3-2

SLOSH PARAMETERS
60% FILL CONDITION
1g ACCELERATION

<u>f</u> (Hz)	<u>Mass</u> <u>lb-sec²/in</u>	<u>Stiffness</u> <u>lb/in</u>	<u>Attach Station (x)</u> <u>in</u>
1.0805	.0008	.0350	22.7
1.1020	.0093	.4458	2.0
1.1284	.0660	3.3176	5.1
1.1384	.0002	.0112	7.3
1.1741	.0230	1.2516	7.7
1.1871	.0007	.0384	-3.8
1.2307	.0041	.2452	3.7
1.2355	.0550	3.3143	3.5
1.2412	.0240	1.4596	1.7
1.2647	.0830	5.2406	1.5
1.2748	.0056	.3592	-1.3
1.2764	.0420	2.7014	0.8
1.3093	.1400	9.4748	2.3
1.4149	.0000	.0001	-94.0

Table 4.3-3

SLOSH PARAMETERS
50% FILL CONDITION
1g ACCELERATION

<u>f</u> <u>(Hz)</u>	<u>Mass</u> <u>1b-sec²/in</u>	<u>Stiffness</u> <u>1b/in</u>	<u>Attach Station (x)</u> <u>in</u>
1.5270	.0003	.0285	-3.6
1.5692	.0062	.6027	10.2
1.5888	.0060	.5979	12.3
1.8572	.0002	.0245	-2.6
1.9030	.0180	2.5735	9.6
2.0489	.0160	2.6518	6.7
2.0725	.0000	.0100	4.7

Table 4.3-4

SLOSH PARAMETERS
40% FILL CONDITION
1g ACCELERATION

<u>f</u> <u>(Hz)</u>	<u>Mass</u> <u>1b-sec²/in</u>	<u>Stiffness</u> <u>1b/in</u>	<u>Attach Station (x)</u> <u>in</u>
1.0594	.0004	.0199	3.3
1.0988	.0001	.0028	-3.3
1.1035	.0210	1.0258	1.3
1.1071	.0008	.0397	5.0
1.1674	.0004	.0200	-2.8
1.2077	.0017	.0978	10.4
1.2184	.0001	.0068	35.6
1.2429	.0002	.0127	-0.3
1.2570	.0005	.0336	28.0
1.2705	.0001	.0072	25.0
A jump was made to higher frequencies			
5.4081	.0003	.3938	37.2
6.0773	.0000	.0032	11.4
7.0748	.0000	.0000	-177.0
8.1944	.0004	1.0516	28.3
8.2019	.0001	.2094	27.5

Table 4.3-5

SLOSH PARAMETERS
20% FILL CONDITION
1g ACCELERATION

<u>f</u> <u>(Hz)</u>	<u>Mass</u> <u>lb-sec²/in</u>	<u>Stiffness</u> <u>lb/in</u>	<u>Attach Station (x)</u> <u>in</u>
1.1170	.0019	.0936	3.7
1.1209	.0060	.2971	3.1
1.1250	.0017	.0084	-0.4
2.6893	.0099	2.8272	1.9
3.3176	.0000	.0162	-16.6

Table 4.3-6

SLOSH PARAMETERS
 80% FILL CONDITION
 $10^{-5}g$ ACCELERATION

<u>f</u> (Hz)	<u>Mass</u> 1b-sec ² /in	<u>Stiffness</u> 1b/in	<u>Attach Station (x)</u> in
0.9374	.0390	1.3469	3.7
0.9803	.0055	.2080	1.0
1.1229	.0006	.0319	17.0
1.1420	.0022	.1151	-2.8
1.1697	.0023	.1235	9.6
1.2470	.0000	.0011	22.0
1.2914	.0012	.0823	9.4
1.3485	.6025	43.2559	7.7
1.3604	.3385	24.7301	7.3
1.3964	.0113	.8699	9.2
1.3967	.0064	.4940	3.4
1.4706	.0011	.0981	2.2
1.5501	.0037	.3546	4.6
1.6660	.0003	.0285	-26.2

Table 4.3-7

SLOSH PARAMETERS
 60% FILL CONDITION
 $10^{-5}g$ ACCELERATION

<u>f</u> (Hz)	<u>Mass</u> 1b-sec ² /in	<u>Stiffness</u> 1b/in	<u>Attach Station (x)</u> in
0.9789	.0006	.0246	18.2
1.0190	.0033	.1334	5.4
1.0461	.0530	2.2743	6.4
1.1590	.0040	.2098	5.9
1.2204	.0001	.0072	8.1

Table 4.3-8

SLOSH PARAMETERS
50% FILL CONDITION
 $10^{-5}g$ ACCELERATION

<u>f</u> (Hz)	<u>Mass</u> 1b-sec ² /in	<u>Stiffness</u> 1b/in	<u>Attach Station (x)</u> in
1.0629	.0008	.0380	-10.9
1.1177	.0180	.8636	-0.4
1.1392	.0043	.2197	-9.9
1.1759	.0095	.5195	0.7
1.1766	.0030	.1620	-2.8
1.2032	.0094	.5393	0.3
1.2247	.0000	.0002	17.9
1.2859	.0030	.1972	-2.4
1.3917	.0000	.0039	5.0
1.3942	.0020	.1569	-2.6
1.3949	.0027	.2103	3.8
1.4356	.0030	.2431	-2.0
1.5645	.0003	.0320	-10.6
1.5788	.0022	.2118	2.8
1.6007	.0001	.0066	-22.0

Table 4.3-9

SLOSH PARAMETERS
40% FILL CONDITION
 $10^{-5}g$ ACCELERATION

<u>f</u> (Hz)	<u>Mass</u> 1b-sec ² /in	<u>Stiffness</u> 1b/in	<u>Attach Station (x)</u> in
1.0036	.0038	.1519	-6.3
1.0191	.0001	.0021	-39.0
1.0238	.0006	.0212	2.9
1.0384	.0026	.1099	-8.7
1.0660	.0015	.0685	1.4
1.0973	.0010	.0475	17.7
1.1022	.0036	.1745	13.3
1.1441	.0001	.0078	42.9
1.1496	.0004	.0200	12.1
1.1772	.0045	.2457	2.9
1.1823	.0220	1.2059	9.4
1.2267	.0000	.0001	10.3
1.2270	.0008	.0471	0.7
1.2514	.0250	1.5727	3.6
1.2555	.0027	.1698	20.1
1.2879	.0060	.3929	4.7
1.3154	.0008	.0548	-0.8
1.3403	.0001	.0070	-3.5
1.3754	.0004	.0306	-0.5

Table 4.3-10

SLOSH PARAMETERS
 20% FILL CONDITION
 $10^{-5}g$ ACCELERATION

<u>f</u> (Hz)	<u>Mass</u> lb-sec/in	<u>Stiffness</u> lb/in	<u>Attach Station (x)</u> in
1.0284	.0008	.0314	-4.9
1.0661	.0030	.1363	-2.6
1.1009	.0000	.0003	-2.07
1.1260	.0000	.0006	55.9
1.1440	.0012	.0615	2.0
1.1624	.0003	.0154	23.5
1.1703	.0017	.0939	1.1
1.1933	.0004	.0235	-9.7
1.2512	.0000	.0025	-48.5
1.3078	.0000	.0023	7.5
1.3175	.0052	.3592	-1.5
1.3381	.0019	.1320	5.2
1.3401	.0011	.0746	-10.1
1.3642	.0007	.0510	-29.8
1.4709	.0001	.0048	-67.8

5. CONCLUSIONS AND COMMENTS

It is felt that a significant contribution was made in this study to the state of the art in finite element fluid analysis. As with all new investigative analytical studies, review of the work performed reveals that although significant advances were made, several items that should have been studied further or have been performed in a different manner if time had permitted.

Several different approaches were developed and programmed to calculate the static equilibrium shape of the fluid/bladder system. Most of them did not perform satisfactorily due to the numerical or convergence problems. Finally, being limited by the time available, a two-dimensional representation - an infinite channel - was chosen for the representation of the equilibrium shape as the only approach showing a reasonably good convergence. A future effort should review this approach critically and perhaps extend it to a 3-dimensional representation.

A second item that should be investigated is the use of double precision in the computer programs, particularly for the calculation of vibration modal properties. This study pointed up the possible need for double precision because a bladder with small modulus of elasticity was used in the analysis. Thus, clear separation of the fluid slosh modes from fluid circulation modes was clouded. In addition, use of low shift values, λ_s , (see Section 2.2.1) sometimes resulted in failure to decompose the dynamical matrix $[K] - \lambda_s [M]$ due to singularity. This is obviously a computer accuracy problem.

The data generator computer subroutine, used to calculate joint X, Y, Z locations, degree of freedom values, Euler angles, and finite element joint numbers has some limitations which should be removed in future studies. One of these limitations is the requirement for vertical radial cuts. This was used to minimize user input but it became obvious later that odd shaped elements resulted. A more general data generator should be coded to allow more user control on the shape of the elements.

The representation of the lateral slosh by a spring mass system has been achieved on the assumption that the modes are completely uncoupled. A detailed investigation of slosh equivalent modeling techniques should be pursued in this direction in a follow-on effort.

REFERENCES

1. "Modal Analysis of Structures by an Iterative Rayleigh-Ritz Technique", by J. R. Admire
2. "Advanced Engineering Mathematics", by C. R. Wylie, Jr., McGraw-Hill, New York, 1960
3. "Linear Strain Membrane Triangle Element", by W. A. Benfield and C. S. Bodley, MMC, DM 169
4. "Triangle Elements in Plate Elements", by C. P. Bazely, Y. K. Cheung, B. M. Irons, and O. C. Zienkiewicz, AFFDL-TR-66-80, November 1966

BIBLIOGRAPHY

1. "Sloshing Liquids in Axisymmetric Ellipsoidal Tanks," by Jasti V. Rattayya, AIAA paper no. 65-114, AIAA 2nd Aerospace Sciences Meeting, New York, N.Y./January 25-27, 1965.
2. "Analytical Representation of Lateral Sloshing by Equivalent Mechanical Models," by Franklin T. Dodge, The Dynamic Behavior of Liquids in Moving Containers, edited by H. Norman Abramson, NASA SP-106.
3. "Slosh Testing of a Spherical Mercury Propellant Tank With Positive-Expulsion Diaphragm," by R. G. Ross, Jr. and J. R. Womack, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, July 15, 1973.
4. "Experimentally Determined Pendulum Analogy of Liquid Sloshing in Spherical and Oblate-Spheroidal Tanks," by I. E. Summer, Lewis Research Center, Cleveland, Ohio. NASA TN D-2737.
5. "Low Gravity Liquid-Vapor Interface Shapes in Axisymmetric Containers and a Computer Solution," by L. J. Hastings and R. Rutherford, III, George C. Marshall Space Flight Center, Alabama, NASA TM X-53790.
6. "Liquid Oscillation Frequencies in Tilted Cylinders for Five Cross-Section Shapes," by S.Thornton and Frank Bugg, Aero-Astroynamics Laboratory, October 1, 1971, NASA TM X-64618.
7. "An Evaluation of the Low G Propellant Behavior of a Space Vehicle During Waiting Orbit," by R. S. Ryan and H. Buchanan, Aero-Astroynamics Laboratory, NASA TM X-53476.

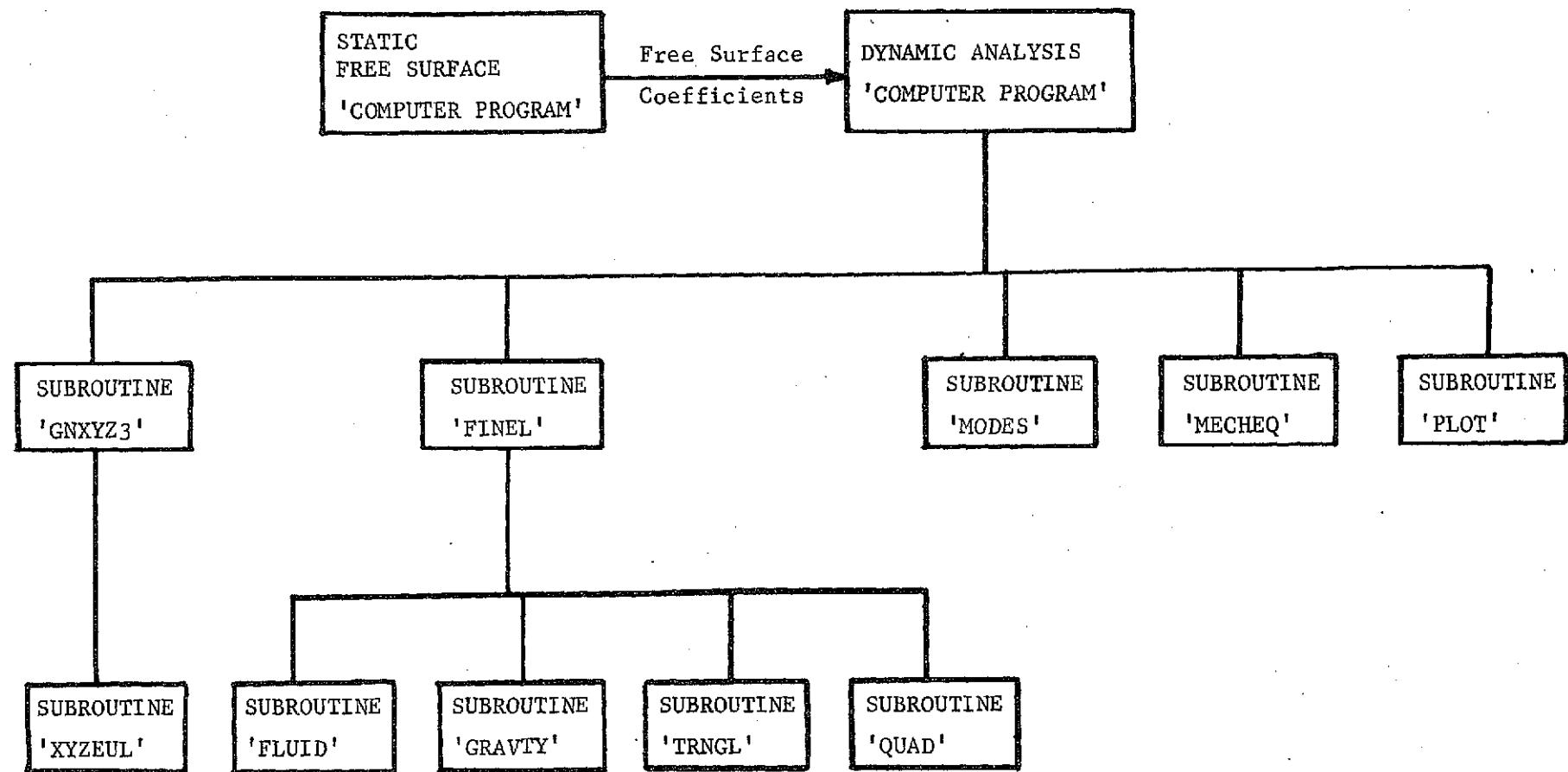
8. "Finite Element Methods in Stress Analysis," edited by Ivar Holand and Kolbein Bell, published by TAPIR, the Technical University of Norway, Trondheim-Norway, 1972.
9. "Introduction to the Finite Element Method," by C. S. Desai and John F. Abel, published by Van Nostrand Reinhold Company.

7. APPENDIX

The analysis of the system is accomplished in two main steps. The first is the static free surface definition and the second is the dynamic analysis. The dynamic analysis program includes the automatic generation of joint X, Y, Z values, degree of freedom numbers, Euler angles, element joint numbers, calculation of mode shapes, and frequencies, mechanical equivalent slosh mass and plots. A computer program has been developed for these steps. A schematic flow chart is shown here for the system analysis steps. The listing of the static free surface computer program is given in Appendix A-1 and that of Dynamic analysis computer program in Appendix A-2. The important parameters input to the programs and subroutines are explained in Appendices B-1 and B-2, respectively, using typical input listings.

A brief summary of important subroutine functions are presented in the following pages.

Brief Summary:



Summary of Programs:

STATIC FREE SURFACE	obtains the static free surface
DYNAMIC ANALYSIS	obtains the dynamic characteristics of the system (frequencies of mode shapes): and the mechanical equivalent

Summary of Subroutines (used in the DYNAMIC ANALYSIS Program):

GNXYZ3	input data for 'FINEL'
XYZEUL	automatic generation of input for 'FINEL'
•	
FINEL	generates mass and stiffness matrices
FLUID	generates mass and stiffness for fluid only
GRAVITY	generates gravity contribution to stiffness matrix
TRNGL	generates mass and stiffness for non-fluid triangular elements
QUAD	generates mass and stiffness for non-fluid quadrilateral elements
MODES	obtains frequencies and mode shapes
MECHEQ	obtains mechanical equivalent for the sloshing fluid and bladder
PLOT	plots the mode shapes for the mid-plane

APPENDIX - A1
STATIC FREE SURFACE PROGRAM

```

PROGRAM SEPS (INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT)          004840
IMPLICIT DOUBLE PRECISION (A-H,O-Z)                         004850
C
      DIMENSION F(15), DF00(15,15), C(15,15), RHSV(15), VEC(15),    004860
      1           VEC1(15), OG(15), C11(15,15), C12(15,15), WORK(15,15), 004880
      2           T(15,15), FT(15,15), C1C2(15,15), PLTIND(83), PLTDEP(83) 004890
C-
      EQUIVALENCE (PLTIND(1),WORK(1)), (PLTDEP(1),WORK(84))        004900
C-
      E X T E R N A L  FT1,FT2,F13,FT4,FT5,FT6,FT7            004910
C
      COMMON/BLK1/ E,TH,RHO,G,P,R,NO                          004920
      C O M M O N  Q(15)                                     004930
C
      1001 FORMAT(15.3D17.8)                                    004940
      2001 FORMAT(1H1,15(/),25X,*MAXIMUM ITERATION LIMIT REACHED*)
      2002 FORMAT (1X,5D20.10)                                 004950
C
      DATA KQ/15/                                         005000
      DATA A/0.0D00/, R/8.0D00/, EPS/1.0D-3/, E/200.0000/   005010
      DATA TH/0.06000/, RHO/0.0013000/, B/8.0D00/             005020
      DATA IN/16/, NC/4/                                     005040
C-
C- *****
C- VARIABLES-
C- IN = NO. OF INTERVALS FOR NUMERICAL INTEGRATION.       005050
C- A = LOWER LIMIT                                         005070
C- R = UPPER LIMIT                                         005080
C- NO = ORDER OF THE POLYNOMIAL                           005090
C- TH = THICKNESS OF THE BLADDER.                         005100
C- E = YOUNGS MODULUS FOR THE BLADDER MATERIAL.          005110
C- RHO = MASS DENSITY OF THE FLUID.                        005120
C- G = ACCELERATION DUE GRAVITY.                         005130
C- P = ULLAGE PRESSURE.                                    005140
C- R = RADIUS OF THE BARREL.                            005150
C- NC = NO. OF CONSTRAINT EQUATIONS.                     005160
C- UV = ULLAGE VOLUME.                                    005170
C- EPS = EPSILON TO COMPARE DU(I)-S.                   005180
C- *****
C-
C- -----
C-
C- INPUTS-
C- NO,G,P,PCT      (15.3D17.8)
C- -----
C-
      10 CALL START                                         005190
      READ(5,1001) NO,G,P,PCT                           005200
      CALL ZERO(DF00,K0,KQ,KQ)                         005210
      CALL ZERO(C,KQ,KQ,KQ)                           005220
      CALL ZERO(VEC,KQ,1,KQ)                           005230
      PT = 2.0 * ASIN(1.0)                           005240
      005250

```

```

ARCVAL = 1.00 * PI * R / 2.0          005290
UV=0.5*(1.0-PCT)*PI*R**2            005300
VEC (NC) = UV                        005310
N1 = NO+1                            005320
N2 = NO + 2                          005330
C-  GENERATE THE MATRIX (C)0.        005340
C-  FIRST ROW                         005350
    C(1,1)=1.0                         005360
    DO 130 J=1,NO                      005370
130  C(1,J+1) = B**J                005380
C-  SECOND ROW                       005390
    C(2,1)=0.0                         005400
    C(2,2)=1.0                         005410
    DO 135 J=2,NO                      005420
135  C(2,J+1) = B** (J-1)*FLOAT(J)
C-  THIRD ROW                         005430
    C(3,1)=0.0                         005440
    C(3,2)=1.0                         005450
    DO 140 J=3,NI                      005460
140  C(3,J)=0.0                      005470
C-  FOURTH ROW                       005480
    DO 145 J=1,NI                      005490
145  C(4,J) = (B**J) / FLOAT(J)
C
C      CALL WRITE (C,NC,N1,2HCO,KQ)    005500
C-  PARTITION (C) TO MAKE (C11) AND (C12) 005510
C
    DO 150 I=1,NC                      005520
    DO 150 J=1,NC                      005530
150  C11(I,J)=C(I,J)                005540
    NC1 = NC + 1                      005550
    NINC=N1-NC                        005560
    DO 155 I=1,NC                      005570
    DO 155 J=1,NINC                   005580
155  C12(I,J)=C(I,J+NC)             005590
    CALL WRITE (C11,NC,NC,3HC11,KQ)    005600
    CALL WRITE (C12,NC,NINC,3HC12,KQ)  005610
C-  GET (C11) INVERSE               005620
    CALL TNVI(C11,WORK,NC,KQ)         005630
    CALL WRITE (WORK,NC,NC,6HC11INV,KQ)
C
C-  GET (-C11) INVERSE TIMES (C12)   005640
    CALL MULT (WORK,C12,C1C2,NC,NC,NINC,KQ,KQ)
    CALL ZERO (T,KQ,KQ,KQ)           005650
    DO 160 I=1,4                      005660
    DO 160 J=1,NINC                   005670
160  T(T,J)=-C1C2(I,J)              005680
    DO 165 I=NC1,NI                   005690
    INC1=I-NC                        005700
165  T(T,INC1)=1.0                  005710
    CALL WRITE (T,NI,NINC,4HTRAN,KQ)  005720
C-  FORM MATRIX (C11) INVERSE TIMES THE RIGHT HAND SIDE 005730
                                         005740
                                         005750
                                         005760
                                         005770
                                         005780

```

```

CALL ZERO (Q,KQ,I,KQ) 005790
CALL MULT (WORK,VEC,Q,NC,NC,I*KQ+KQ) 005800
Q(N2) = SIMPS(A,B,IN,FT7,TDUM,IDUM)/ARCVAL - 1.0 005810
CALL WRITE (Q,N2,I,6H0(I)-I,KQ) 005820
C 005830
C 005840
C- LOOP TO DETERMINE Q(I)=S. 005850
MM = 0 005860
100 CONTINUE 005870
CALL MULT(C,Q,WORK,NC,N1,I,KQ+KQ) 005880
CALL WRITE(WORK,NC,I,6HCONSTR,KQ) 005890
WRITE(6,2000) MM 005900
2000 FORMAT(* ITERATION NO. *,I3) 005910
MM = MM +1 005920
TF (MM.GT.50) GO TO 190
C- FOR THE FIRST N+1,F=S. 005940
DO 110 L = 1+N1 005950
110 F(L) = SIMPS (A,B,IN,FT1,L,1DUM) 005960
C- FOR THE LAST N+2 ND. F 005970
CALL WRITE (F,N1+1,2HFN,KQ) 005980
C- CALCULATE (DFDQ) MATRIX. 005990
DO 115 I=1,N1 006000
DO 115 J=1,1 006010
115 DFDQ(I,J) = SIMPS(A,B,IN,FT3,I,J) 006020
DO 120 I = 1,N1 006030
DO 120 J = 1,1 006040
120 DFDQ(J,I) = DFDQ(I,J) 006050
CALL WRITE (DFDQ,N1,N1+4HDFDQ,KQ) 006060
C 006070
C- GET (T) TRANSPOSE AND MULTIPLY 006080
DO 170 I=1,N1 006090
DO 170 J=1,N1NC 006100
170 TT(J,I) = F(I,J) 006110
C 006120
CALL MULT (TT,F,N1NC,N1,I,KQ,KQ) 006130
C- OBTAIN (T) TRANSPOSE X (DFDQ) X (T) 006140
CALL BTABA (DFDQ,T,N1,N1NC,KQ,KQ) 006150
CALL WRITE (DFDQ,N1NC,N1NC,6HTTDFT,KQ) 006160
CALL INV1(DFDQ,TT,N1NC,KQ) 006170
CALL MULT (TT,F,N1NC,N1NC,I,KQ+KQ) 006180
C- OBTAIN DELTA-A2 006190
CALL WRITE (F,N1NC,I,3H0N2,KQ) 006200
C 006210
C- GET DELTA-A. 006220
C 006230
CALL MULT (T+F,DQ,N1,N1NC,I*KQ,KQ) 006240
C 006250
CALL WRITE (DQ,N1,I,2H0N2,KQ) 006260
C 006270
C- GET Q(PRESENT) = Q(PREVIOUS)+DQ 006280
C 006290
DO 175 I=1,N1 006300

```

```

175 Q(I) = Q(I) - DQ(I)          006310
      Q(N2) = SIMPS(A,B,N,F,T7,1DUM,1DUM)/ARCVAL = 1.0 006320
C
      CALL WRITE (9,N2+1,5HNQI-S,KW) 006330
      WRTTE (6,2002) (Q(I),I=1,N2) 006340
C
      DO 180 LL=1,N1               006350
      IF (DABS(DQ(LL)).GT.EPS) GO TO 100 006360
180 CONTINUE                      006370
C- PLT FREE SURFACE SHAPE        006380
      PLTIND(1) = R                006390
      PLTDEP(1) = 2*R              006400
      PLTIND(83) = 0.0              006410
      PLTDEP(83) = 0.0              006420
      DEL = R / 80.0               006430
      X = A + DEL                006450
      DO 185 I = 2,81              006470
      X = X + DEL                006480
      PLTIND(I) = X               006490
      PLTDEP(I) = Z(X)            006500
185 CONTINUE                      006510
      PLTIND(82) = R               006520
      PLTDEP(82) = Z(R)             006530
      CALL WRITE(PLTIND,83,2,6HSHAPE ,83) 006540
      GO TO 195                  006550
190 WRTTE (6,2001)                006560
195 CONTINUE                      006570
      GO TO 10                  006580
      END                         006590
      DOUBLEPRECISION FUNCTION SIMPS (A,B,N,F,NP,INT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z) 006600
C
C- THIS FUNCTION SUPPLIES THE NUMERICALLY INTEGRATED VALUE OF THE 006610
C- INTEGRAND. 006620
C
      COMMON/BLK1/ E,TH,RHO,G,P,E,NO
      COMMON Q(15)                 006630
C
C- A=LOWER LIMIT                 006640
C- B=UPPER LIMIT                 006650
C- N=NO. OF INTERVALS             006660
C- F=FUNCTION                      006670
C- NP=GENERALIZED COORDINATE NUMBER (AS TO WHICH ONE IT IS) 006680
C
C- INITIALIZE PARAMETERS          006690
      TWOH=(B-A)/N                006700
      H = TWOH/2.0                 006710
      SUMEND=0.0                   006720
      SUMMID = 0.0                  006730
C
C- TWOH=INTERVAL                  006740
C- H=HALF INTERVAL                 006750

```

```

C-      SUMMEND=SUM OF F(XI), I BEING EVEN EXCEPTING I=2N          006790
C-      SUMMID=SUM OF F(XI), I BEING ODD.                         006800
C                                         006810
C-      EVALUATE SUMEND AND SUMMID.                                006820
DO 1 K = 1,N                                                       006830
X=A+FLOAT(K-1)*TWOH                                             006840
SUMEND = SUMEND + F(X,NP,INT)                                     006850
1 SUMMID = SUMMID + F(X+H,NP,INT)                                 006860
C                                         006870
C-      RETURN ESTIMATED VALUE OF THE INTEGRAL.                  006880
STMP5 = (2.0*SUMEND+4.0*SUMMID-F(A,NP,INT)+F(B,NP,INT))*H/3.0  006890
C                                         006900
RETURN                                                               006910
END                                                               006920
DOUBLEPRECISION FUNCTION FT1(X,NP,IOU)                           006930
IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               006931
C-      THIS FUNCTION DEFINES THE INTEGRAND FOR F-S FOR DISCRETE VALUES OF 006940
C      FOP DIFFERENT EQUATION NUMBERS CORRESPONDING TO DIFFERENT Q-S OR A 006950
C      NP = EQUATION NUMBER                                         006960
C.                                         006970
COMMON/BLK1/  E,TH,RHO,G,P,R,NO                                006980
COMMON Q(15)                                                       006990
C                                         007000
NP = NO + 2                                                       007010
C                                         007020
IF (X .EQ. 0.0 .OR. NP .LE. 2) GO TO 5                         007030
TFLOT1 = -FLOAT((NP-1)*(NP-2)) * X***(NP-3)                   007040
GO TO 10                                                       007050
5 TFLOT1 = 0.0                                                   007060
10 CONTINUE                                                       007070
IF (X .EQ. 0.0 .OR. NP .LE. 1) GO TO 15                         007080
TFLOT2 = -FLOAT(NP-1) * X***(NP-2)                               007090
GO TO 20                                                       007100
15 TFLOT2 = 0.0                                                   007110
20 CONTINUE                                                       007120
C                                         007130
TERM1=((E*TH**3)/12.0)*(((1.0+ZP(X)**2)**(-3.50))           007140
1     *((((1.0+ZP(X)**2)*(ZDP(X))* TFLOT1))                  007150
2     ~(( 2.50)*(ZDP(X)**2)*(ZP(X))* TFLOT2))                007160
C                                         007170
TERM2=(E*TH**Q(N2)**2*0.5)*(((1.0+ZP(X)**2)**(-0.5))         007180
1     *(ZP(X))* TFLOT2)                                         007190
C                                         007200
IF (X .EQ. 0.0) GO TO 25                                         007210
TERM3= (RHO*G)*Z(X)*(-(X***(NP-1)))                          007220
C                                         007230
TERM4 = P*(X***(NP-1))                                         007240
C                                         007250
GO TO 30                                                       007260
25 CONTINUE                                                       007270
TERM3 = 0.0                                                       007280
TERM4 = 0.0                                                       007290
C                                         007300

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

30 FT1=TERM1+TERM2+TERM3+TERM4          007310
C
C      RETURN                               007320
C      END                                  007330
C      DOUBLEPRECISION FUNCTION FT2(X,NP,IDU) 007340
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)   007350
C-     THIS FUNCTION DEFINES THE INTEGRAND FOR DISCRETE X FOR (NO+2) EQUA 007360
C
C      COMMON/BLK1/  E,TH,RHO,G,P,R,NO        007370
C      C O M M O N  Q(15)
C
C      NP = NO + 2                          007380
C
C      TTERM=(E*TH*Q(N2))*((1.0+ZP(X)**2)**(0.5)) 007390
C
C      FTP = TERM                           007400
C
C      RETURN                               007410
C      END                                  007420
C      DOUBLEPRECISION FUNCTION Z(X)        007430
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)   007440
C-     THIS FUNCTION DEFINES THE FUNCTION Z(X) AT DISCRETE X. 007450
C
C      COMMON/BLK1/  E,TH,RHO,G,P,R,NO        007460
C      C O M M O N  Q(15)
C
C      IF (R .EQ. X) GO TO 5                007470
C      Z = (R**2-X**2)**0.5+R-W(X)          007480
C      GO TO 10                            007490
C      5 Z = R - W(X)                      007491
C      10 CONTINUE                         007500
C
C      RETURN                               007510
C      END                                  007520
C      DOUBLEPRECISION FUNCTION W(X)        007530
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)   007540
C-     THIS FUNCTION DEFINES W(X) AT DISCRETE VALUE OF X. 007550
C
C      COMMON/BLK1/  E,TH,RHO,G,P,R,NO        007560
C      C O M M O N  Q(15)
C
C      N1 = NO + 1                          007570
C      W1 = 0.0                             007580
C      IF (X .EQ. 0.0) GO TO 15            007590
C      DO 10 I=1,N1
C      10 W1=W1+Q(I)*X***(I-1)           007600
C      GO TO 20                            007610
C      15 W1= W1 + Q(1)                   007620
C      20 CONTINUE                         007630
C      W = W1                            007640
C
C      RETURN                               007650
C

```

```

END 007800
DOUBLEPRECISION FUNCTION ZP(X) 007810
IMPLICIT DOUBLE PRECISION ( A-H,O-Z) 007811
THIS FUNCTION DEFINES (Z) AT DISCRETE X. 007820
007830
C- COMMON/BLK1/ E,TH,RHO,G,P,R,NO 007840
C COMMON Q(15) 007850
C
N1 = NO + 1 007860
TERM1=0.0 007870
IF (X .EQ. 0.0) GO TO 15 007880
IF (R .EQ. X) GO TO 20 007890
DO 10 I=2,N1 007900
10 TERM1= TERM1+ (Q(I)*X**(I-2))*FLOAT(I-1) 007920
TERM1=TERM1 + X*(R**2-X**2)**(-0.5) 007930
GO TO 25 007940
15 TERM1 = Q(2) 007950
GO TO 25 007960
20 CONTINUE 007970
DO 30 I=2,N1 007980
30 TERM1 = TERM1 + (Q(I) * X**(I-2)) * FLOAT(I-1) 007990
25 CONTINUE 008000
C TERM = -TERM1 008010
C ZP=TERM 008020
C
RETURN 008030
END 008040
008050
DOUBLEPRECISION FUNCTION ZDP(X) 008060
IMPLICIT DOUBLE PRECISION ( A-H,O-Z) 008070
008080
C THIS FUNCTION DEFINES (Z) FOR DISCRETE X. 008090
008100
COMMON/BLK1/ E,TH,RHO,G,P,R,NO 008110
COMMON Q(15) 008120
N1 = NO + 1 008130
C TERM1=0.0 008140
C
IF (X .EQ. 0.0) GO TO 15 008150
IF (R .EQ. X) GO TO 20 008160
DO 10 I=3,N1 008170
10 TERM1=TERM1+ (Q(I)*X**((I-3))*FLOAT((I-1)*(I-2))) 008180
TERM1= TERM1+ (R**2-X**2)**(-0.5) 008190
1 +X**2*(R**2-X**2)**(-1.5) 008200
1 GO TO 25 008210
15 TERM1 = Z.0 * Q(3) + (1.0 / R) 008220
1 GO TO 25 008230
20 CONTINUE 008240
DO 30 I=3,N1 008250
20 CONTINUE 008260
30 CONTINUE 008270

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

30 TERM1=TERM1+(0(I)*X**((I-3))*FLOAT((I-1)*(I-2)))          008280
25 CONTINUE
C
C      TERM = -TERM1
C
C      ZDP=TERM
C
C      RETURN
C      END
C      DOUBLE PRECISION FUNCTION FT3(X,NP,M)
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      THIS FUNCTION DEFINES THE INTEGRAND FOR OF(K)/UD(M) FOR DISCRETE X
C      K=1,2,---(NO+1) AND M=1,2,---,(NO+1). IN THE MATRIX K IS THE ROW
C      AND M IS THE COLUMN.
C
C      COMMON/BLK1/ E,TH,RHO,G,P,R,F0
C      C O M M O N 3(15)
C
C      IF (X .EQ. 0.0 .OR. NP .LE. 2) GO TO 5
C      TFLOT1 = -FLOAT((NP-1)*(NP-2)) * X**((NP-3))
C      GO TO 10
5     TFLOT1 = 0.0
10    CONTINUE
      IF (X .EQ. 0.0 .OR. NP .LE. 1) GO TO 15
      TFLOT2 = -FLOAT(NP-1) * X**((NP-2))
      GO TO 20
15    TFLOT2 = 0.0
20    CONTINUE
C
C      NP = NO + 2
C
C      A1 = 1.0 + ZP(X)**2
C      B1 = ZDP(X)
C      C1 = TFLOT1
C      A2 = ZDP(X)**2
C      B2 = ZP(X)
C      C2 = 2.50 * TFLOT2
C      A3 = ((1.0+ZP(X)**2)**(-0.50))
C      B3 = ZP(X)
C      C3 = TFLOT2
C      D = (1.0 + ZP(X)**2) **(-3.50)
C      A1AM = 2.0 * ZP(X) * PZP(X,M)
C      B1AM = PZDP(X,M)
C      A2AM = 2.0 * ZDP(X) * PZDP(X,M)
C      B2AM = PZP(X,M)
C      A3AM = -ZP(X) * ((1.0+ZP(X)**2)**(-1.50)) * PZP(X,M)
C      B3AM = PZP(X,M)
C      DAM = (-7.0) * ((1.0 + ZP(X)**2) ** (-4.50)) * ZP(X) * PZP(X,M)
C
C      TFPML = ((E*TH**3)/12.0) *
1        (0 * ((A1*C1*B1AM) + (B1*C1*A1AM) - (A2*C2*B2AM) -
2        (B2*C2*A2AM)) + (A1*B1*C1) + A2*B2*C2) * DAM)
C

```

```

TERM2 = (0.50*E*TH*Q(N2)**2) +
1      (A3*C3*B3AM + B3*C3*A3AM) 008800
008810
008820
C      TF(X.EQ.0.0) GO TO 25 008830
TERM3=(RH0*G)*(PZ(X,M)*(-(X** (NP-1)))) 008840
GO TO 30 008850
25 TERM3 = 0.0 008860
008870
C      30 TERM= TERM1+TERM2+TERM3 008880
008890
C      FT3=TERM 008900
008910
C      RETURN 008920
END 008930
DOUBLEPRECISION FUNCTION FT4(X,NP,M) 008940
IMPLICIT DOUBLE PRECISION (A-H,O-Z) 008941
C      THIS FUNCTION DEFINES INTEGRAND FOR DF(K)/DQ(M) FOR DISCRETE X AND 008950
C      K=1,2,---(NO+1) AND M=NO+2 008960
008970
C      COMMON/BLK1/ E,TH,RH0,G,P,R,NO 008980
COMMON/BLK1/ E,TH,RH0,G,P,R,NO 008990
009000
C      NO = NO + 2 009010
IF (X.EQ.0.0 .OR. NP.LE.1) GO TO 15 009020
TFLOT2 = -FLUAT(NP-1)*X** (NP-2) 009030
GO TO 20 009040
15 TFLOT2 = 0.0 009050
20 CONTINUE 009060
009070
C      TERM=(E*TH*Q(N2))*((1.0+ZP(X)**2)**(-0.5)*ZP(X) * TFLOT2) 009080
009090
C      FT4 = TERM 009100
009110
C      RETURN 009120
END 009130
009140
DOUBLEPRECISION FUNCTION FT5(X,NP,M) 009150
IMPLICIT DOUBLE PRECISION (A-H,O-Z) 009151
C      THIS FUNCTION DEFINES INTEGRAND FOR DF(K)/DQ(M) FOR DISCRETE X AND 009160
C      K=NO+2 AND M=1,2,---NO+1. 009170
009180
C      COMMON/BLK1/ E,TH,RH0,G,P,R,NO 009190
COMMON/BLK1/ E,TH,RH0,G,P,R,NO 009200
009210
C      NO = NO + 2 009220
TERM=(E*TH*Q(N2))*((0.5)*(1.0+ZP(X)**2)**(-0.5)
1      *(2.0*ZP(X)*PZP(X,M))) 009230
009240
009250
C      FT5 = TERM 009260
009270
C      RETURN 009280
END 009290

```

```

DOUBLEPRECISION FUNCTION F16(X,NP,M)          009300
IMPLICIT DOUBLE PRECISION (A-H,O-Z)          009301
C- THIS FUNCTION DEFINES INTEGRAND FOR DF(K)/DQ(M) FOR DISCRETE X AND 009310
C- K=N0+2 AND M=N0+2                         009320
C                                         009330
COMMON/BLK1/ E,TH,RHO,S,P,R,N0             009340
C M M O N 0(15)                            009350
C                                         009360
TERM=(E*TH)*(1.0+ZP(X)**2)**(0.5)           009370
C                                         009380
ETA=TERM                                     009390
C                                         009400
RETURN                                       009410
END                                         009420
DOUBLEPRECISION FUNCTION PZDP(X,M)           009430
IMPLICIT DOUBLE PRECISION (A-H,O-Z)           009431
C- THIS FUNCTION CALCULATES THE VALUE OF Q(Z) OR PZDP FOR DISCRETE X. 009440
IF (X .EQ. 0.0) GO TO 10                      009450
IF (M .GT. 2) GO TO 5                         009460
TERM = 0.0                                     009470
GO TO 15                                      009480
5 CONTINUE                                     009490
TERM = -FLOAT((M-1)*(M-2)) * X** (M-3)       009500
GO TO 15                                      009510
10 CONTINUE                                     009520
TERM = 0.0                                     009530
IF (M .EQ. 3) TERM = -2.0                      009540
15 CONTINUE                                     009550
PZDP = TERM                                    009560
C                                         009570
RETURN                                       009580
END                                         009590
DOUBLEPRECISION FUNCTION PZP(X,M)            009600
IMPLICIT DOUBLE PRECISION (A-H,O-Z)           009601
C- THIS FUNCTION CALCULATES THE VALUE OF Q (Z) OR PZP(X) FOR DISCRETE 009610
IF (X .EQ. 0.0) GO TO 10                      009620
IF (M .GT. 1) GO TO 5                         009630
TERM = 0.0                                     009640
GO TO 15                                      009650
5 CONTINUE                                     009660
TERM = -FLOAT(M-1) * X** (M-2)                 009670
GO TO 15                                      009680
10 CONTINUE                                     009690
TERM = 0.0                                     009700
IF (M .EQ. 2) TERM = -1.0                      009710
15 CONTINUE                                     009720
PZP = TERM                                    009730
C                                         009740
RETURN                                       009750
END                                         009760
DOUBLEPRECISION FUNCTION PZ(X,M)             009770
IMPLICIT DOUBLE PRECISION (A-H,O-Z)           009771

```

```

C- THIS FUNCTION CALCULATES THE VALUE OF 0(Z)ORPZ(X) FOR DISCRETE X. 009780
  IF (X .EQ. 0.0) GO TO 5 009790
  TFRM = -(X**(M-1)) 009800
  GO TO 10 009810
5 CONTINUE 009820
  TFRM = 0.0 009830
  IF (M .EQ. 1) TERM = -1.0 009840
10 CONTINUE 009850
  PZ = TERM 009860
C
C RETURN 009870
END 009880
DOUBLEPRECISION FUNCTION FT7(X, IDUM, JDUM) 009890
IMPLICIT DOUBLE PRECISION (A-H,O-Z) 009900
C
C- FUNCTION TO EVALUATE ARC LENGTH OF DEFORMED BLADDER 009910
C
FT7 = DSQRT(1.0 + ZP(X)*ZP(X)) 009920
RETURN 009930
END 009940
SUBROUTINE PAGEHD 009950
COMMON/LSTART/IRUNNO, IDATE, NPAGE, UNAME(3), TITLE1(12), TITLE2(12)
DATA NIT, NOT/5,6/ 009960
C
2001 FORMAT (9H1RUN NO. ,A6,42X,5HDATE ,A6/
  *55X,7HRUN BY ,3A6/102X,A9,12H CLOCK TIME/10X,12A6,19X,F10.3,
  *12H SEC. CPTIME/10X,12A6)
C
CALL TIME (DTIME)
CALL SECOND (CP)
WRITE (NOT,2001) IRUNNO, IDATE, UNAME, DTIME,
*           TITLE1, CP, TITLE2
C
RETURN
END
SUBROUTINE MULTB (A,BZ,NRA,NRR,NCB,KA+KBZ) 000100
IMPLICIT DOUBLE PRECISION (A-H,O-Z) 000110
DIMENSION A(KA,1), BZ(KBZ,1) 000120
COMMON /LWRKV1/ W(40) 000130
C
C MATRTX MULTIPLICATION. A * R = Z. 000140
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN R. 000150
C BZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE 000160
C LARGER OF B OR Z. 000170
C CALLS FORMA SUBROUTINE ZZBOMB. 000180
C THE MAXIMUM SIZE IS 000190
C   NRR = 40 000200
C DEVELOPED BY CARL BUDLEY. JANUARY 1965. 000210
C LAST REVISION BY R L WOHLEN. JULY 1972. 000220
C
C SURROUTINE ARGUMENTS 000230
C A = TINPUT MATRIX. SIZE(NRA,NRR). 000240
C

```

```

C   BZ = INPUT MATRIX. SIZE(NRA,NCB).          000270
C   = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).      000280
C   NRA = INPUT NUMBER OF ROWS OF MATRICES A,Z.  000290
C   NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A. MAX=40. 000300
C   NCB = INPUT NUMBER OF COLS OF MATRICES B,Z.  000310
C   KA = INPUT ROW DIMENSION OF A IN CALLING PROGRAM. 000320
C   KBZ = INPUT ROW DIMENSION OF BZ IN CALLING PROGRAM. 000330
C                                         000340
C                                         NERROR=1 000350
C   IF (NRB.GT.40 .OR. NRA.GT.KBZ .OR. NRB.GT.KBZ) GO TO 999 000360
C                                         000370
C   DO 40 J=1,NCB 000380
C   DO 20 K=1,NRB 000390
20  W(K) = BZ(K,J) 000400
C   DO 40 I=1+NRA 000410
C   S = 0. 000420
C   DO 30 K=1,NRB 000430
30  S = S + A(I,K)*W(K) 000440
40  BZ(I,J) = S 000450
C   RRETURN 000460
C                                         000470
999 CALL ZZBOMB (6HMULTB ,NERROR) 000480
C   END 000490
C   SUBROUTINE ZZBOMB (SUBNAM,NERROR) 000500
C   DIMENSION DFMSSG(8) 000510
C   DATA NIT,NOT/5,6/ 000520
C                                         000530
C   CONTROL A COMPUTER RUN AFTER AN ERROR MESSAGE HAS BEEN ENCOUNTERED 000540
C   IN ANY OF THE FORMA SUBROUTINES. 000550
C   ON THE CDC 6000 SERIES COMPUTER THIS INVOLVES ... 000560
C   (1) PRINT ERROR MESSAGE, INCLUDING SUBNAM AND NERROR, IN PRINTOUT. 000570
C   (2) PRINT ERROR MESSAGE, INCLUDING SUBNAM AND NERROR, IN DAYFILE. 000580
C   (3) CALL TO NON-EXISTANT ROUTINE TO CAUSE ABNORMAL 000590
C   STOP AND TRANSFER TO THE EXIT CARD. 000600
C   CODED BY RL WOHLEN. SEPTEMBER 1970. 000610
C   LAST REVISION BY R HRUDA, JAN 1974. 000620
C                                         000630
C   SUBROUTINE ARGUMENTS 000640
C   SUBNAM = INPUT SUBROUTINE NAME IN WHICH ERROR OCCURRED. 000650
C   NERROR = INPUT ERROR NUMBER FROM SUBROUTINE WHERE ERROR OCCURRED. 000660
C                                         000670
3001 FORMAT (1H1) 000680
3002 FORMAT (19H ZZBOMB - ROUTINE (,A6,11H), NERROR (,I3,1H)) 000690
C                                         000700
C   WRITE (NOT,3001) 000710
C   WRITE (NOT,3002) SUBNAM,NERROR 000720
C   ENCODE (40,3002,DFMSSG) SUBNAM,NERROR 000730
C   CALL REMARK (DFMSSG) 000740
C   CALL ARNORML 000750
C                                         000760
C   END 000770
C   SUBROUTINE BTABA (AZ,B,NRB,NCB,KAZ,KB) 000810

```

```

IMPLICIT DOUBLE PRECISION (A-H,O-Z)          000820
DIMENSION AZ(KAZ,1), B(KB,1)                  000830
COMMON / LWRKVI / W(40)                      000840
000850
C TRIPLE MATRIX PRODUCT. B(TRANSPOSE) * A * B = Z. 000860
C A MUST BE SYMMETRIC TO GET CORRECT ANSWER. 000870
C Z WILL BE SYMMETRIC. UPPER HALF CALCULATED, REFLECTED TO LOWER HALF. 000880
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A. 000890
C AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE 000900
C LARGER OF A OR Z. 000910
C CALLS FORMA SUBROUTINE ZZBOMR. 000920
C THE MAXIMUM SIZES ARE 000930
C      NRR = 40 000940
C      NCR = 40 000950
C DEVELOPED BY W A BENFIELD. MAY 1972. 000960
C LAST REVISION BY R A PHILIPPUS. JUNE 1972. 000970
C 000980
C   SURROUNTING ARGUMENTS 000990
C AZ = INPUT INNER MATRIX. SIZE(NRB,NRB). 001000
C      = OUTPUT RESULT MATRIX. SIZE(NCB,NCB). 001010
C B = INPUT OUTER MATRIX. SIZE(NRR,NCR). 001020
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, SIZE OF MATRIX A. MAX=40. 001030
C NCR = INPUT NUMBER OF COLS OF MATRIX B, SIZE OF MATRIX Z. MAX=40. 001040
C KAZ = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM. 001050
C KR = INPUT ROW DIMENSION OF B IN CALLING PROGRAM. 001060
C 001070
C           NERROR=1 001080
IF (NRB.GT.40 .OR. NCR.GT.40 .OR. NRB.GT.KAZ .OR. NCB.GT.KAZ) 001090
* GO TO 999 001100
C 001110
DO 20 I=1,NRB 001120
DO 5 K=1,NRB 001130
5 W(K) = AZ(I,K) 001140
DO 20 J=1,NCB 001150
S = 0.0 001160
DO 10 K=1,NRB 001170
10 S = S + W(K)*B(K,J) 001180
20 AZ(I,J) = S 001190
C 001200
DO 30 J=1,NCB 001210
DO 25 I=1,J 001220
W(I) = 0.0 001230
DO 25 K=1,NRB 001240
25 W(I) = W(I)+B(K,I)*AZ(K,J) 001250
DO 30 I=1,J 001260
AZ(T,J) = W(I) 001270
30 AZ(J,I) = W(I) 001280
RETURN 001290
C 001300
999 CALL ZZBOMR (6HB(A,B,A ,NERRH) 001310
END 001320
SUBROUTINE INV1 (A,Z,N,KR) 001330

```

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)          001340
      DIMENSION A(1), Z(1)                         001350
      COMMON /LWRKV1/ G(20), DETR(20)              001360
      COMMON /LWRKV2/ IX(20), B(20)                001370
      DATA NIT,NOT/5,6/                            001380

C
C  MATRIX INVERSION (A**-1 = Z). BORDERING METHOD.          001390
C  THE DETERMINANT RATIO DET(I+1) / DET(I) IS PRINTED. DET(I) IS THE 001400
C  DETERMINANT OF THE FIRST I BY I SUB-MATRIX OF A.          001410
C  THE INVERSION CHECK Z*A IS CALCULATED AND PRINTED.        001420
C  MATRICES A,Z MAY SHARE SAME CORE LOCATIONS. (Z*A CHECK IS INVALID). 001430
C  CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.                  001440
C  THE MAXIMUM SIZE IS                                001450
C      N = 20                                         001460
C  DEVELOPED BY BOB DILLON. FEBRUARY 1965.            001470
C  LAST REVISION BY J ERNST, OCT 1973.             001480
C
C      SUBROUTINE ARGUMENTS                           001490
C  A = TINPUT MATRIX TO BE INVERTED. SIZE(N,N).          001500
C  Z = OUTPUT RESULT MATRIX. SIZE(N,N).               001510
C  N = TINPUT SIZE OF MATRICES A,Z. MAX=20.           001520
C  KP = TINPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM. 001530
C
2000 FORMAT (// 10X,1U(7X,1H(,I2,1H)))          001540
2001 FORMAT (// 10X,4SHSUBROUTINE INV1 HAS CALCULATED THE DATA BELOW 001550
      *      // 10X,44HTHE DETERMINANT RATIOS DET(I+1) / DET(I) ARE 001560
      *      // (13X,10D11.3))                      001570
2002 FORMAT (///10X,37HTHE (A**-1)* (A) INVERSION CHECK GIVES 001580
      *      // 10X,25HTHE DIAGONAL ELEMENTS ARE // (13X,8D14.6)) 001590
2003 FORMAT (// 10X,35HTHE MAXIMUM OFF-DIAGONAL ELEMENT IS 001600
      *      D11.3, 2X, 4HAT ( 13, 1H, 13, 1H ) ) 001610
C
      IF (N .GT. 20) GO TO 999                         NERROR=1 001620
C
      DO 160 IX(I) = I                                 001630
160   IX(I) = I
C  INVERT FIRST NON-ZERO ELEMENT IN FIRST COLUMN.       001640
      DO 190 I=1,N
      IF (A(I) .NE. 0.) GO TO 220
190   CONTINUE
      GO TO 999                                     NERROR=2 001650
C
C  START INVERSION WITH ROW 1.                         001660
220   DFTR(1) = A(I)
      Z(I) = 1. / A(I)
      IF (N .EQ. 1) RETURN
C
      IX(I) = 1
      IX(I) = I
C  BORDERING LOOP.                                    001670

```

```

DO 630 L=2,N          001860
  K = L                001870
  L1 = L - 1           001880
250 S = 0.              001890
                         001900
                         001910
                         001920
                         001930
                         001940
                         001950
                         001960
                         001970
                         001980
                         001990
                         002000
                         002010
                         002020
                         002030
                         002040
                         002050
                         002060
                         002070
                         002080
                         002090
                         002100
                         002110
                         002120
                         002130
                         002140
                         002150
                         002160
                         002170
                         002180
                         002190
                         002200
                         002210
                         002220
                         002230
                         002240
                         002250
                         002260
                         002270
                         002280
                         002290
                         002300
                         002310
                         002320
                         002330
                         002340
                         002350
                         002360
                         002370

  MIXL = KR * (IX(L) - 1)
  LL = IX(L) + MIXL
DO 450 I=1,L1          001900
  MIXI = KR * (IX(I) - 1)
  LI = IX(L) + MIXI
B(I) = 0.                001910
G(I) = 0.                001920
DO 440 J=1,L1          001930
  MIXJ = KR * (IX(J) - 1)
  IJ = IX(I) + MIXJ
  JL = IX(J) + MIXL
  R(T) = B(I) - Z(IJ)* A(JL)
  JI = IX(J) + MIXI
  LJ = IX(L) + MIXJ
440 G(I) = G(I) - A(LJ)* Z(JI)
450 S = S + A(LI)* B(I)
  AL = A(LL)+ S
  IF (A(LL) .EQ. 0.) GO TO 480
  ALBAR = DABS (AL / A(LL))
  GO TO 490
480 ALBAR = DABS (AL)
490 IF (ALBAR .GE. .1D-6) GO TO 550
C
C INTERCHANGE ROWS AND COLUMNS.
  K = K + 1
  IF (K .GT. N) GO TO 540
  IX_L = IX(L)
  IX(L) = IX(K)
  IX(K) = IX_L
  GO TO 250
540 IF (ALBAR .GE. .1D-8) GO TO 550
                         NERROR=3
  GO TO 999
C
550 Z(IL)= 1. / AL      002210
  DFTR(L) = AL
  DO 570 I=1,L1          002220
    IL = IX(I) + MIXL
    LI = IX(L) + KR * (IX(I) - 1)
    Z(IL)= B(I) * Z(LL)
    Z(LI)= G(I) * Z(LL)
    DO 570 J=1,L1          002230
      TJ = IX(I) + KR * (IX(J) - 1)
      Z(TJ)= Z(IJ)+ G(J) * Z(IL)
570 Z(IL)= Z(IL)+ G(I) * Z(IL)
630 CONTINUE
C
C COMPUTE INVERSION CHECK Z*A.
  XOFF = 0.0

```

```

DO 720 I=1,N          002380
DO 710 J=1,N          002390
X = 0.0               002400
KJA = KR * (J-1)      002410
DO 703 K=1,N          002420
IK = I + KR*(K-1)     002430
KJ = K + KJA          002440
703 X = X + Z(IK) * A(KJ) 002450
IF (I .NE. J) GO TO 705 002460
G(I) = X              002470
GO TO 710              002480
705 IF (DABS(X) .LT. DABS(XOFF)) GO TO 710 002490
XOFF = X              002500
IOFF = I              002510
JOFF = J              002520
710 CONTINUE           002530
720 CONTINUE           002540
C                         002550
C   PRINT THE DETERMINANT RATIO AND INVERSION CHECK. 002560
C   CALL PAGEHD          002570
WRTTE (NOT,2000) (JC, JC=1+10) 002580
WRTTE (NOT,2001) (DETR(I), I=1,N) 002590
WRTTE (NOT,2002) ( G (I), I=1,N) 002600
WPTTE (NOT,2003) XOFF,IOFF,JOFF 002610
RETURN                 002620
C                         002630
C   999 CALL ZZBOMB (SHINV1 ,NERROR) 002640
END                     002650
SURROUNTING MULT (A,B,Z,NRA,NRB,NCH,KRA,KRB) 002660
IMPLICIT DOUBLE PRECISION (A-H,O-Z) 002670
DIMENSION A(KRA,1), B(KRB,1), Z(KRA,1) 002680
C                         002690
C   MATRIX MULTIPLICATION. A * B = Z. 002700
C   DEVELOPED BY R L WOHLEN. FEBRUARY 1965. 002710
C   LAST REVISION BY R L WOHLEN. JULY 1972. 002720
C                         002730
C   SURROUNTING ARGUMENTS 002740
C   A = INPUT MATRIX. SIZE(NRA,NRS). 002750
C   B = INPUT MATRIX. SIZE(NRB,NCH). 002760
C   Z = OUTPUT RESULT MATRIX. SIZE(NRA,NCB). 002770
C   NRA = INPUT NUMBER OF ROWS OF MATRICES A,Z. 002780
C   Z = OUTPUT RESULT MATRIX. SIZE(NRA,NCB). 002770
C   NRA = INPUT NUMBER OF ROWS OF MATRIX A, COLS OF MATRIX B. 002790
C   NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A. 002800
C   NCB = INPUT NUMBER OF COLS OF MATRICES B,Z. 002810
C   KRA = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM. 002820
C   KRB = INPUT ROW DIMENSION OF B IN CALLING PROGRAM. 002830
C
DO 20 I=1,NRA          002840
DO 20 J=1,NCH          002850
S = 0.                 002860
DO 10 K=1,NRB          002870
10 S = S + A(I,K)*B(K,J) 002880
20 Z(I,J) = S          002890

```

```

RETURN 002900
END 002910
SUBROUTINE MULTA (AZ,B,NRA,NRB,NCB,KAZ,KB) 002920
IMPLICIT DOUBLE PRECISION (A-H,O-Z) 002930
DIMENSION AZ(KAZ,1), B(KB,1) 002940
COMMON / LWRKVI / W(40) 002950
002960
C 002970
C MATRIX MULTIPLICATION. A * B = Z. 002980
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A. 002990
C AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE 003000
C LARGER OF A OR Z. 003010
C CALLS FORMA SUBROUTINE ZZB0MR. 003020
C THE MAXIMUM SIZE IS 003030
C NRP = 500 003040
C DEVELOPED BY C S BODLEY. JANUARY 1965. 003050
C LAST REVISION BY R F HRUDA. JUNE 1972. 003060
C 003070
C SUBROUTINE ARGUMENTS 003080
C AZ = INPUT MATRIX. SIZE(NRA,NRB). 003090
C = OUTPUT RESULT MATRIX. SIZE(NRA,NCB). 003100
C B = INPUT MATRIX. SIZE(NRB,NCB) 003110
C NRA = INPUT NUMBER OF ROWS OF MATRICES A,Z. 003120
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A. MAX=500. 003130
C NCB = INPUT NUMBER OF COLS OF MATRICES B,Z. 003140
C KAZ = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM. 003150
C KR = INPUT ROW DIMENSION OF B IN CALLING PROGRAM. 003160
C 003170
C IF (NRP .GT. 500) GO TO 999 NERROR=1 003180
C 003190
C DO 40 I=1,NRA 003200
C DO 20 K=1,NRB 003210
20 W(K) = AZ(I,K) 003220
DO 40 J=1,NCB 003230
S = 0.0 003240
DO 30 K=1,NRB 003250
30 S = S + W(K)*B(K,J) 003260
40 AZ(I,J) = S 003270
RETURN 003280
003290
C 999 CALL ZZB0MB (6HMULTA ,NERROR) 003300
END 003310
SUBROUTINE WRITE (A,NR,NC,INAME,KR) 003320
IMPLICIT DOUBLE PRECISION (A-H,O-Z) 003330
DIMENSION A(KR,1) 003340
DATA NIT,NOT/5,6/ 003350
003360
C WRITE MATRIX OF REAL NUMBERS ON PAPER. 003370
C REQUESTS 123 COLUMN (MINIMUM) PRINTER. 003380
C UP TO 10 DATA FIELDS PER LINE. PRINTS ONLY NON-ZERO FIELD ROWS. 003390
C CALLS FORMA SUBROUTINE PAGEHD. 003400
C CODED BY RL WOHLER. DECEMBER 1968. 003410

```

```

C LAST REVISION BY R HRUDA, NOV 1973.          003420
C
C SURROUNTING ARGUMENTS (ALL INPUT)           003430
C A      = MATRIX TO BE PRINTED. SIZE(NR,NC).   003440
C NR     = NUMBER OF ROWS IN MATRIX A.         003450
C NC     = NUMBER OF COLS IN MATRIX A.         003460
C INAME  = MATRIX IDENTIFICATION. (A6 FORMAT). 003470
C KR     = ROW DIMENSION OF A IN CALLING PROGRAM. 003480
C
2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H) // 003490
*      10X,10(7X,1H( I2,1H))/) 003500
2020 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H ) 003510
*      3X, 9HCONTINUED //10X,10(7X,1H( I2,1H))/) 003520
2030 FORMAT (1X,2I5,2X, 10D11.3) 003530
2040 FORMAT (14H0END OF WRITE.) 003540
C
C PULL UP A NEW PAGE FOR MATRIX AND PRINT MATRIX NAME. 003550
    CALL PAGEHD 003560
    WRITE (NOT,2010) INAME,NR,NC,(L,L=1,10) 003570
    NLINE = 0 003580
C
    DO 60 I=1,NR 003590
    NZERO = 0 003600
    JS = 1 003610
10  JF = JS+9 003620
    IF (JE .GT. NC) JE=NC 003630
C SEE IF ELEMENTS ARE ZERO. 003640
    DO 20 J=JS,JE 003650
        IF (A(I,J) .NE. 0.) GO TO 30 003660
20  CONTINUE 003670
    GO TO 40 003680
30  NLINE = NLINE+1 003690
    IF (NLINE .LE. 44) GO TO 35 003700
    CALL PAGEHD 003710
    WRITE (NOT,2020) INAME,NR,NC,(L,L=1,10) 003720
    NLINE = 1 003730
35  WRITE (NOT,2030) I,JS,(A(I,J), J=JS,JF) 003740
    NZERO = 1 003750
40  IF (JE .EQ. NC) GO TO 50 003760
    JS = JS+10 003770
    GO TO 10 003780
C SKIP A SPACE BETWEEN EACH ROW IF THERE ARE MORE THAN 10 COLUMNS 003790
C AND SOMETHING HAS BEEN WRITTEN. 003800
50  IF (NC.LE.10 .OR. NZERO.EQ.0 .OR. I.EQ.NR) GO TO 60 003810
    NLINE = NLINE+1 003820
    WRITE (NOT,2030) 003830
60  CONTINUE 003840
C
    WRITE (NOT,2040) 003850
    RETURN 003860
    END 003870
003880
003890
003900
003910
003920

```

ORIGINAL PAGE
OF POOR QUALITY

```

SUBROUTINE ZERO(Z,NR,NC,KR)          003930
IMPLICIT DOUBLE PRECISION (A-H,O-Z)  003940
DIMENSION Z(KR,1)                   003950
C                                     003960
C GENERATE A MATRIX OF ZEROES.       003970
C CODED BY RL WOHLEN. FEB 1965.      003980
C                                     003990
C SUBROUTINE ARGUMENTS              004000
C Z = OUTPUT MATRIX GENERATED. SIZE(NR,NC). 004010
C NR = TINPUT NUMBER OF ROWS IN MATRIX Z. 004020
C NC = TINPUT NUMBER OF COLS IN MATRIX Z. 004030
C KR = INPUT ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM. 004040
C                                     004050
DO 10 I=1,NR                      004060
DO 10 J=1,NC                      004070
10 Z(I,J) = 0.0                    004080
RETURN                            004090
END                                004100
SUBROUTINE START                  004110
DIMENSION MONTHN(12),MONTHL(12)    004120
COMMON /LSTART/IRUNNO,IDATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12) 004130
DATA NIT,NOT/S,6/                 004140
DATA MONTHN/2H01,2H02,2H03,2H04,2H05,2H06, 004150
*           2H07,2H08,2H09,2H10,2H11,2H12/. 004160
*   MONTHL/2HJA,2HFE,2HMR,2HAP,2HMY,2HJN, 004170
*           2HJL,2HAU,2HSE,2HOC,2HNO,2HDE/ 004180
C                                     004190
C READS INPUT CARD 1 FOR IRUNNO, UNAME. 004200
C CHECKS IRUNNO FOR STOP (I.E. IF IRUNNO = STOP, PROGRAM WILL 004210
C BE STOPPED). YOU SHOULD HAVE A STOP CARD (THE WORD STOP PUNCHED 004220
C STARTTING IN COLUMN 1) AFTER YOUR REGULAR DATA DECK. 004230
C IF IRUNNO IS NOT EQUAL TO STOP, THE SUBROUTINE CONTINUES AS FOLLOWS. 004240
C READS INPUT CARD 2 FOR TITLE1. 004250
C READS INPUT CARD 3 FOR TITLE2. 004260
C SETS NPAGE = 0. 004270
C INTERROGATES COMPUTER TO DEFINE DATE AS AN A6. 004280
C INTERROGATES MACHINE FOR THE TIME OF DAY AND THE OPTIME 004290
C AND PRINTS THESE ITEMS ON A SHEET OF THE OUTPUT EVERY 004300
C TIME THIS ROUTINE IS CALLED. 004310
C                                     004320
C     INPUT ORDER                  004330
C IRUNNO,UNAME        FORMAT (A6, 4X 3A6) 004340
C TITLFI             FORMAT (12A6)        004350
C TITLF?             FORMAT (12A6)        004360
C                                     004370
C     DEFINITIONS                 004380
C IRUNNO = RUN NUMBER. (A6 FORMAT) 004390
C IDATE = DATE. (A6 FORMAT).    004400
C NPAGE = PAGE NUMBER.         004410
C UNAME = USERS NAME. (3A6 FORMAT) 004420
C TITLE1 = FIRST TITLE. (12A6 FORMAT) 004430
C TITLE2 = SECOND TITLE. (12A6 FORMAT) 004440
C                                     004450
C DEDICATED TO G. MOROSOW.      004460

```

C CALLS CDC-6000-SERIES COMPUTER STANDARD ROUTINES DATE, SECOND, TIME.004470
C CALLS MARTIN MARIETTA SPECIAL ROUTINE PPTIM. 004480
C CODED FOR YOUR CONVENIENCE BY YOUR FRIENDLY METHODS GROUP. DEC 1968.004490
C LAST REVISION BY R HRUDA, MAY 1974. 004500
C 004510
1001 FORMAT (A6, 4X 3A6) 004520
1002 FORMAT (12A6) 004530
2002 FORMAT (1H1 6(/) 55X 10HTIME SHEET / 38X 45(1H-) // 004540
* 38X 30HCURRENT TIME OF DAY IN H,M,S = A10 // 004550
* 38X 26HTOTAL CPTIME USED TO NOW = F10.3, 9H SECONDS. / 004560
* 38X 26HTOTAL PPTIME USED TO NOW = I6.4X, 9H SECONDS.) 004570
2003 FORMAT (36H1END OF INPUT DATA HAS BEEN REACHED.) 004580
5001 FORMAT (3(1X,A2)) 004590
5002 FORMAT (3A2) 004600
C 004610
CALL TIME (DTIME)
CALL SECOND (CTIME)
WRTE (NUT,2002) DTIME,CTIME,IPTIME 004630
CALL DATE (IDATE)
DECODE (9,5001,DATE) IM, ID, IY 004650
DO 20 I=1,12 004660
IF (IM.EQ.MONTHN(I)) GO TO 30 004670
20 CONTINUE 004680
30 IM = MONTHL(I)
ENCODE (6,5002,DATE) ID, IM, IY 004690
C 004700
READ (NIT,1001) IRUNNO,UNAME 004710
IF (IRUNNO .NE. 4HSTOP) GO TO 10 004720
WRTE (NUT,2003)
STOP 004730
C 004740
10 READ (NIT,1002) TITLE1
READ (NIT,1002) TITLE2 004750
NPAGE = 0 004760
RETURN 004770
END 004780
004790
004800
004810
004820
004830

APPENDIX - A2
DYNAMIC ANALYSIS PROGRAM

SIRISHPOOOOO=F2+MAIN

```

1   C
2   C OVERLAY MAIN PROGRAM TO CALCULATE SHUTTLE EXTERNAL TANK SLOSH MODES.
3   C DEVELOPED BY W BENFIELD, C BUDLEY, R PHILIPPUS, R WOHLEN, JULY 1973.
4   C LAST REVISION BY R A PHILIPPUS. JULY 1974.
5   C
6   C *****
7   C INPUT DATA READ IN THIS PROGRAM.
8   C 10 CALL START
9   C CALL COMENT
10  C IFINIT,TAPEID           FORMAT (2A6)
11  C 15 IOPT                 FORMAT (A6)
12  C IF (IOPT .EQ. 6HSTART ) GO TO 10
13  C IF (IOPT .EQ. 6HGNXYZ ) CALL GNXYZ22 (SEE SUBRT FOR INPUT)
14  C IF (IOPT .EQ. 6HFINEL ) CALL FINEL  (INPUT DATA FROM GNXYZ22)
15  C IF (IOPT .EQ. 6HXTRAMK) CALL PXTTRA (SEE SUBRT FOR INPUT)
16  C IF (IOPT .EQ. 6HMODES ) CALL GNIMD  (SEE SUBRT FOR INPUT)
17  C *IF (IOPT .EQ. 6HMODES ) CALL OYMODE (SEE SUBRT FOR INPUT)
18  C IF (IOPT .EQ. 6HREDUCE) CALL REDUCE (SEE SUBRT FOR INPUT)
19  C IF (IOPT .EQ. 6HSAVEMK) CALL SAVE   (SEE SUBRT FOR INPUT)
20  C IF (IOPT .EQ. 6HSUBSTR) CALL SUBSTR (SEE SUBRT FOR INPUT)
21  C IF (IOPT .EQ. 6HPLOT  ) CALL SPLT1  (SEE SUBRT FOR INPUT)
22  C GO TO 15
23  C
24  COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
25  COMMON / BTAPEF / NUTMF,NUTKF,NUT1F,NUT2F,NUT3F
26  COMMON / BTAPET / NUTMT,NUTKT,NUTIT,NUT2T,NUT3T,NUT4T,NUTST
27  COMMON / BTAPEM / NUTMM,NUTKM,NUTTM,NUTPM,NUTFM,NUT1M,NUT2M,NUT3M,
28  *          NUT4M,NUT5M,NUT6M,NUT7M
29  COMMON / BTAPED / NUTKD,NUTLD,NUTDD,NUT1D,NUT2D,NUT3D,NUT4D
30  COMMON / BTAPEB / NUTKB,NUTBB,NUTPB,NUT1B,NUT2B,NUT3B,NUT4B,
31  *          NUT5B,NUT6B,NUT7B
32  COMMON / BTAPEC / NUTNC,NUTKC,NUTTC,NUT1C,NUT2C,NUT3C,NUT4C,NUT5C,
33  *          NUT6C
34  COMMON / BTAPER / NUTMR,NUTKR,NUTTR,NUT1R,NUT2R,NUT3R,NUT4R,NUT5R,
35  *          NUT6R
36  COMMON / BTAPES / NUTMS,NUTKS,NUTTS
37  COMMON / BTAPEP / NUTMP,NUTKP,NUTTP,NUT1P,NUT2P,NUT3P,NUT4P,NUT5P
38  COMMON / BTAPEA / NUTPA,NUTFA,NUT1A,NUT2A,NUT3A,NUT4A,NUT5A,
39  *          NUT6A,NUT7A
40  COMMON / RESTAP / NR5VTI
41  COMMON / RTRANS / ITRAN
42  C
43  DATA NIT,NOT/5+6/
44  C DEFINE READ,WRITE TAPE UNITS FOR ALL OVERLAYS.
45  DATA NUTEL,NUTXYZ /
46  *          29,    30 /
47  DATA NUTLT,NUTST,NUTMX,NUTKX,NUTBX /
48  *          31,    1,    2,    26,   27 /
49  C DEFINE BUFFER IN,OUT TAPE UNITS FOR FINEL0.
50  DATA NUTMF,NUTKF,NUT1F,NUT2F,NUT3F/
51  *          21,    22,   11,    12,   13/
52  C DEFINE BUFFER IN,OUT TAPE UNITS FOR EXTRA M,K OVERLAY.
53  DATA NUTMT,NUTKT,NUTIT,NUT2T,NUT3T,NUT4T,NUTST /
54  *          21,    22,   11,    12,   13,   14,   15 /
55  C DEFINE BUFFER IN,OUT TAPE UNITS FOR YMODE2.
56  DATA NUTMM,NUTKM,NUTPM,NUTTM,NUTFM,NUT1M,NUT2M,NUT3M,NUT4M,NUT5M/

```

```

57      *      21,   22,   23,   24,   25,   11,   12,   13,   14,   15/
58      DATA NUT6M/
59      *      16/
60      DATA NUT7M/
61      *      17/
62 C  DEFINE BUFFER IN,OUT TAPE UNITS FOR STATIC DEFLECTION CALCULATIONS.
63      DATA NUTKD,NUTLD,NUTDD,NUT1D,NUT2D,NUT3D,NUT4D /
64      *      22,   21,   23,   11,   12,   13,   14 /
65 C  DEFINE BUFFER IN,OUT TAPE UNITS FOR BUCKLING LOAD CALCULATIONS.
66      DATA NUTKB,NUTBB,NUTPB,NUT1B,NUT2B,NUT3B,NUT4B,NUT5B,NUT6B,NUT7B/
67      *      22,   21,   23,   11,   12,   13,   14,   15,   16,   17/
68 C  DEFINE BUFFER IN,OUT TAPE UNITS FOR CONSTANT VOLUME FLUID ELEMENT
69 C  CALCULATIONS.
70      DATA NUTMC,NUTKC,NUTTC,NUT1C,NUT2C,NUT3C,NUT4C,NUTSC,NUT6C/
71      *      21,   22,   24,   11,   12,   13,   14,   15,   16/
72 C  DEFINE BUFFER IN,OUT TAPE UNITS FOR REDUCING CALCULATIONS.
73      DATA NUTMR,NUTKR,NUTTR,NUTIR,NUT2R,NUT3R,NUT4R,NUTSR,NUT6R/
74      *      21,   22,   24,   11,   12,   13,   14,   15,   16/
75 C  DEFINE BUFFER IN,OUT TAPE UNITS FOR SUBSTRUCTURE CALCULATIONS.
76      DATA NUTMP,NUTKP,NUTTP,NUTIP,NUT2P,NUT3P,NUT4P,NUT5P/
77      *      21,   22,   24,   11,   12,   13,   14,   15/
78 C  DEFINE BUFFER IN,OUT TAPE UNITS FOR PLOTTING.
79      DATA NUTPA,NUTFA,NUTIA,NUTZA,NUT3A,NUT4A,NUTSA,NUT6A,NUT7A/
80      *      23,   24,   11,   12,   13,   14,   15,   16,   17/
81 C  DEFINE BUFFER IN,OUT TAPE UNITS FOR SAVEMK.
82      DATA NUTMS,NUTKS,NUTTS/
83      *      21,   22,   24/
84 C  DEFINE RESERVE TAPES.
85      DATA NRSVT1           / 28   /
86 C
87 1001 FORMAT (12A6)
88 1010 FORMAT (10X 15)
89 C
90      IF (NRSVT1 .GT. 0) REWIND NRSVT1
91 10 CALL START
92      CALL COMENT
93      READ (INIT,1001) IFINITE,TAPEID
94      IF (IFINITE .EQ. 6HINIT1L) CALL INTAPE (NRSVT1,TAPEID)
95      IFTRAN = 0
96 15 READ (INIT,10G11) IOPT
97      IF (IOPT .EQ. 6HSTART ) GO TO 10
98      IF (IOPT .EQ. 6HGNXYZ ) GO TO 18
99      IF (IOPT .EQ. 6HFINEL ) GO TO 20
100     IF (IOPT .EQ. 6HXTRAMK) GO TO 30
101     IF (IOPT .EQ. 6HMODES ) GO TO 40
102     IF (IOPT .EQ. 6HMECHEQ) GO TO 70
103     IF (IOPT .EQ. 6HREDUCE) GO TO 80
104     IF (IOPT .EQ. 6HSAVEMK) GO TO 90
105     IF (IOPT .EQ. 6HSUBSTR) GO TO 100
106     IF (IOPT .EQ. 6HPLOT ) GO TO 110
107
108     GO TO 999
109 18 CALL GNXYZZ
110     GO TO 15
111 20 CALL OFINEL
112     GO TO 15
113 30 CALL PXTRA

```

ORIGINAL PAGE IS
OF POOR QUALITY

NERROR=1

```
114      GO TO 15
115      40 CALL GNJMD
116      CALL DYMODE
117      GO TO 15
118      70 CALL MECHEQ
119      GO TO 15
120      80 CALL REDUCE
121      GO TO 15
122      90 CALL SAVE
123      GO TO 15
124      100 CALL SUBSTR
125      GO TO 15
126      110 CALL OPLT10
127      GO TO 15
128
129      999 CALL ZZBOMB (6HPSL0SH,NERROR)
130      END
```

MPRT F1-BASIC

```

SIRISHPO00000*F1+BASIC
1      SUBROUTINE BASIC  (XYZ,JDOF,EUL,KRX,KCX,KRJ,KCJ,KRE,KCE,
2      *                      NUTEL,NUTXYZ)
3      C
4      C  SUBROUTINE TO READ BASIC FINEL DATA FROM CARD INPUT AND WRITE ON
5      C  NUTEL AND NUTXYZ.
6      C  DEVELOPED BY WA BENFIELD, APRIL 1974.
7      C
8      DIMENSION XYZ(KRX,1), JDOF(KRJ,1), EUL(KRE,1)
9      DIMENSION IDATA (14)
10     C
11     DATA NJT,NOT / 5,6 /
12 1001 FORMAT (13A6,A2)
13  REWIND NUTEL
14  C  READ JOINT XYZ COORDINATE MATRIX.
15 10 CALL READ  (XYZ,NJ,NCX,KRX,KCX)
16  C  READ JOINT DEGREE OF FREEDOM MATRIX.
17  CALL READIM (JDOF,NRJ,NCJ,KRJ,KCJ)
18  C  READ JOINT EULER ANGLES.
19  CALL READ  (EUL,NRE,NCE,KRE,KCE)
20  C  READ ONE CARD WITH NAMEL FOR FINEL.
21  C  READ DATA CARDS FOR AXIAL OR BAR OR TRNGL, ETC.
22 15 READ (NIT ,1001) IDATA
23  WRITE (NUTEL,1001) IDATA
24  IF (IDATA(1),EQ,6HRETURN) GO TO 100
25  GO TO 15
26  C
27 100 REWIND NUTEL
28  REWIND NUTXYZ
29  WRITE (NUTXYZ) NJ,NCX,NRJ,NCJ,NRE,NCE
30  WRITE (NUTXYZ) ((JDOF(I,J),I=1,NRJ),J=1,NCJ)
31  WRITE (NUTXYZ) ((XYZ(I,J),I=1, NJ),J=1,NCX)
32  WRITE (NUTXYZ) (( EUL(I,J),I=1,NRE),J=1,NCE)
33  REWIND NUTXYZ
34  RETURN
35
36  C
END

```

BPRT F1+GNIMD

```
SIRISHP00000*F1.GNIMD
1      SUBROUTINE GNIMD
2      C
3      C ***** INPUT DATA READ IN THIS PROGRAM *****
4      C
5      C      CALL YREAD (INITIAL DISPL MODES)
6      C      RETURN
7      C
8      COMMON /BTAPEM/ NUTM,NUTK,NUTTR,NUTZ,NUT1,NUT2,NUT3,NUT4,NUT5,
9      *          NUT6,NUT7
10     DIMENSION V(12000), LV(12000)
11     DATA KV / 12000 /
12     CALL YREAD (NUTZ, V,LV,KV,NUT1)
13     RETURN
14     END
```

NPRT F1.GNXYZZ

```

SIRISHPOOOOO*F1*GNXYZ2
1      SUBROUTINE GNXYZ2
2
3      C MAIN PROGRAM TO GENERATE (XYZ), (JUUF), AND (EUL); AND STORE MATRICES
4      C ON UTILITY TAPE.
5      C DEVELOPED BY W BENFIELD, C BODLEY, R PHILIPPUS, R WOHLER. JULY 1973.
6      C LAST REVISION BY R A PHILIPPUS. JUNE 1974.
7      C
8      C ***** INPUT DATA READ IN THIS PROGRAM. *****
9
10     C
11     C       IOPT           FORMAT (A6)
12     C       IF (IOPT .EQ. 6HBASIC) CALL READ  (XYZ,NJ,3)
13     C       *                   CALL READIM (JUUF,NJ,6)
14     C       *                   CALL READ  (EUL,NJ,3)
15     C       *                   CALL FINEL  (SEE SUBRT FOR INPUT)
16     C       IF (IOPT .EQ. 6HXYZEUL) CALL DATGEN (SEE SUBRT FOR INPUT)
17     C       RETURN
18
19     COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
20     DATA NIT,NOT / 5,6 /
21
22     1001 FORMAT (13A6,A2)
23
24     REWIND NUTEL
25     READ (NIT,1001) IOPT
26     IF (IOPT .EQ. 6HBASIC) GO TO 10
27     IF (IOPT .EQ. 6HFLAT P) GO TO 23
28     IF (IOPT .EQ. 6HCNT BM) GO TO 35
29     IF (IOPT .EQ. 6HCYLCOR) GO TO 40
30     IF (IOPT .EQ. 6HCIRCYL) GO TO 50
31     IF (IOPT .EQ. 6HNASTRN) GO TO 60
32     IF (IOPT .EQ. 6HSNAP ) GO TO 70
33     IF (IOPT .EQ. 6HXYZEUL) GO TO 80
34
35     GO TO 999
36
37     10 CALL PBASIC
38     REWIND NUTEL
39     REWIND NUTXYZ
40     RETURN
41
42     20
43     GO TO 999
44     30
45     GO TO 999
46     40
47     GO TO 999
48     50
49     GO TO 999
50     60
51     GO TO 999
52     70
53     GO TO 999
54     80 CALL XYZEU2
55     REWIND NUTEL
56     REWIND NUTXYZ
57     RETURN
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999

```

57 C
58 999 CALL ZZBOMB (6HGNXYZZ,NERROR)
59 END

WPRTR F1-MECH_EQ

SIRISHP00000*F1.MECHEW

```

1      SUBROUTINE MECHEW
2      COMMON /BTAPED/ NUTKD,NUTLD,NUTDD,NUT11,NUT12,NUT13,NUT14
3      DIMENSION V(4000),LV(4000),T(1500,6),TF(600,6),X(405,3),
4      *JD(405,6),JVEC(6),E(405,3),R(6,6),RT(6,6),EM(6,6)
5      DIMENSION TT(6,600),XR(3),RV(6,6),IVEC(6),ITVC(600)
6      DATA KV ,KR ,KJ ,KD ,KX , KBT/
7      *     4000, 600, 405,   6,    3, 1500/
8      EQUIVALENCE(T,V),(T(4001),LV)

9      C  PROGRAM TO CALCULATE EFFECTIVE MASS FOR SLOSH
10     C  DEVELOPED BY P W ABBOTT. DECEMBER 1974.
11     C  FORM RIGID BODY TRANSFORMATION IN GLOBAL
12     C
13     C 2001 FORMAT (//,10X,13HSLUSH MASS = ,E11.4,5X,8HCP(X) = ,E11.4)
14     C
15     CALL READ(X,NR,NC,KJ,KX)
16     READ(5,101) N3,NBLAD
17     101 FORMAT(16I5)
18     C  N3 IS THE LAST NODE WITH ONLY 3 DOF
19     C  NBLAD IS THE DOF NO. AT N3.
20     DO 6 I=1,N3
21     DO 6 J=1,3
22     6 JD(I,J)=J+3*(I-1)
23     N=N3+1
24     DO 7 I=N,NR
25     DO 7 J=1,6
26     7 JD(I,J)=J+3*(I-N)+6
27     CALL WRITIM(JD,NR,6,6HJD0FUL,KJ)
28     CALL READ(XR,NF,NC,I,KX)
29     CALL READIM(JVEC,NRJ,NCJ,I,KD)
30     CALL RBTG1(X,XR,JD,JVEC,I,NR,NRT,NCT,KJ,KBT)
31     CALL WRITE(T,NRT,NCT,6HRGL0BL,KBT)
32
33     C  TRANSFORM TO LOCAL SYSTEM
34     CALL READIM(JD,NR,NG,KJ,KD)
35     CALL READ(E,NR,NC,KJ,KX)
36     DO 100 I=1,6
37     100 IVEC(I)=I
38     L=1
39     M=0
40     MI = 0
41     DO 10 I=1,NR
42     CALL ZERO(JVEC,I,6,1)
43     CALL ZERO(R,6,6,KD)
44     KROT=3
45     IF(I.GT.N3) KROT=6
46     K=0
47     DO 9 J=1,6
48     IF(JD(I,J).LT.0) GO TO 888
49     IF(J.LT.4) XR(J)=E(I,J)
50     IF(JD(I,J).LE.0) GO TO 9
51     IF (I.GT.N3 .AND. JD(I,J).LE.NBLAD) GO TO 9
52     K=K+
53     JVEC(J)=K
54
55
56

```

```

57      MI = MI+1
58      ITVC(MI) = JD(I,J)
59      * CONTINUE
60      CALL EULER(XR,R,KD)
61      IF(KROT.EQ.6) CALL EULER(XR,R(4,4),KD)
62      CALL ZERO (RV,KROT,K,KD)
63      CALL REVADD (1,R,IVEC,JVEC,RT,KROT,KROT,K,KD,KU)
64      CALL TRANS(RV,RT,KROT,K,KD,KD)
65      CALL MULTA(RT,T(L,1),K,KROT,NCT,KD,KBT)
66      DO 8 IT=1,K
67      DO 8 JT=1,NCT
68      8 TF(M+IT,JT)=RT(IT,JT)
69      M=M+K
70      888 L=L+KROT
71      10 CONTINUE
72      CALL WRITE(TF,M,NCT,6HROTRAN,KR)
73      DO 889 J=1,3
74      889 JVEC(J) = J
75      CALL ZERO (T,M,NCT,KBT)
76      CALL REVADD (1,TF,ITVC,JVEC,T,M,NCT,M,NCT,KR,KBT)
77      CALL WRITE (T,M,NCT,6HCORTRN,KBT)
78      CALL TRANS(T,TT,M,NCT,KBT,KD)
79      C
80      C READ MASS AND MODES AND DO THE REST
81      CALL YREAD (NUT11,V,LV,KV,NUT14)
82      CALL YREAD (NUT12,V,LV,KV,NUT14)
83      CALL YMULT1 (NUT11,NUT12,NUT13,V,LV,KV,NUT14)
84      CALL YSTOD (NUT13,TF,NR,NC,KR,KD,V,LV,KV,NUT14)
85      CALL MULT (TT,TF,R,NCT,NR,NC,KD,KR)
86      DO 15 I=1,NCT
87      DO 15 J=1,NC
88      15 TT(I,J) = RI(I,J)
89      CALL TRANS(TT,TF,NCT,NC,KD,KR)
90      CALL WRITE(TT,NCT,NC,6HTTMPI,KU)
91      CALL PAGEHD
92      DO 20 I=1,NC
93      CALL MULT(TT(I,I),TF(I,I),EM,NCT,I,NCT,KD,KR)
94      C 20 CALL WRITE(EM,NCT,NCT,NAME(4HMASS,I),KD)
95      CPRESS = EM(2,3)/EM(2,2)
96      20 WRITE (6,2001) EM(2,2),CPRESS
97      RETURN
98      END

```

```

SIRISHP00000*FI*OFINEL
1      SUBROUTINE OFINEL
2
3      C MAIN PROGRAM TO READ (XYZ), (JDOF), (EUL) AND CALCULATE (ON OPTION),
4      C ASSEMBLED FINITE ELEMENT MASS, STIFFNESS, LOAD TRANSFORMATION, AND
5      C BUCKLING MATRICES.
6      C CALLS FORMA SUBROUTINES FINEL ,YIN ,YWRITE .
7      C DEVELOPED BY W BENFIELD, C BODLEY, R PHILIPPUS, R WOHLER. JULY 1973.
8      C LAST REVISION BY R A PHILIPPUS. DECEMBER 1974.
9      C
10     COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
11     COMMON / BTAPEF / NUTM,NUTK,NUT1,NUT2,NUT3
12     COMMON / DUMMY / V(12000)
13     COMMON / IDUMMY / LV(12000)
14     COMMON / RESTAP / NRSVT1
15     DIMENSION XYZ(2000,3), JDOF(2000,6), EUL(2000,3)
16     EQUIVALENCE (XYZ,V), (EUL,V(6001)), (JDOF,LV)
17     DATA KRX, KCX, KRJ, KCJ, KRE, KCE, KV /
18     *   2000, 3, 2000, 6, 2000, 3, 12000 /
19     C READ XYZ,JDOF,EUL FROM NUTXYZ CREATED IN PROGRAM DATGEN.
20     REWIND NUTXYZ
21     READ (NUTXYZ) NJ,NCX,NRJ,NCJ,NRE,NCE
22
23     IF (NCX .NE. 3) GO TO 999
24
25     IF (NRJ .NE. NJ .OR. NCJ .NE. 6) GO TO 999
26
27     IF (NRE .NE. NJ .OR. NCE .NE. 3) GO TO 999
28
29     IF ( ( NJ.GT.KRX .OR. NCX.GT.KCX .OR.
30     *      NRJ.GT.KRJ .OR. NCJ.GT.KCJ .OR.
31     *      NRE.GT.KRE .OR. NCE.GT.KCE) ) GO TO 999
32     READ (NUTXYZ) ((JDOF(I,J),I=1,NRJ),J=1,NCJ)
33     READ (NUTXYZ) ((XYZ(I,J),I=1, NJ),J=1,NCX)
34     READ (NUTXYZ) (( EUL(I,J),I=1,NRE),J=1,NCE)
35     CALL WTAPF (XYZ,NJ,NCX,6HXYZ ,KRX,NRSVT1)
36     CALL WTAPF (JDOF,NRJ,NCJ,6HJDOF ,KRJ,NRSVT1)
37     CALL WTAPF ( EUL,NRE,NCE,6HEUL ,KRE,NRSVT1)
38     CALL FINEL (XYZ,JDOF,EUL,NUTEL,NJ,
39     *           NUTM,NUTK,NUTLT,NUTST,NUTBX,
40     *           V,LV,KV,KRX,KRJ,KRE,
41     *           NUTMX,NUTKX,NUT1,NUT2,NUT3)
42     CALL YWRITE (NUTM,4HMASS ,V,LV,KV)
43     CALL YTAPF (NUTM,6HMASS ,V,LV,KV,NRSVT1)
44     CALL YWRITE (NUTK,4HSTIF ,V,LV,KV)
45     CALL LTAPF (NRSVT1)
46     RETURN
47     C
48     999 CALL ZZBOMB (6HUFINEL,NERRR)
49     END

```

```

SIRISHP00000*F1*OPLT10
1      SUBROUTINE OPLT10
2      COMMON /BTAPED/ NUTKD,NUTLD,NUTOD,NUTI,NUTP,NUT3,NUT4
3      DIMENSION XYT(45,3), PLOC(90,20), XP(200), YP(200), XPTYP(90,5),
4      *          DX(45), DY(45), DAL(45), DYL(45), PTITLE(4),
5      *          SCALEM(20), FREQM(20), BIGP(600,6), IVEC(45), JDOF(600,6),
6      *          V(4000), LV(4000)
7      DATA KJ, KP, KM, KPLUT, KBIGP, KBIGM, KV/
8      *        45, 90, 20, 200, 600, 6, 4000/
9      DATA EPS/1.E-15/
10     DATA NIT,NUT/5,6/
11    1001 FORMAT (10X 15,15)
12    2001 FORMAT (10X, F10.4,F10.4)
13    2002 FORMAT (//10X, 9HMODE NO. I3, //)
14    2003 FORMAT (//10X, 9HMODE NO. I3, 5X 7HSCALE = F6.3)
15    5001 FORMAT (5HMODE=I3, 1X 5HFREQ=F7.4, 1X 5HSHT=I4, 1X 4HSCL=F4.3)
16    CALL IDFRMV (12H G I BULTMAN ,12H BIN 5 190 ,6H5 6325 ,
17    *           12HHARDCOPY )
18
19 C   READ IVEC. IN SURFACE JOINT ORDER FROM CENTER OUT. VALUE IS ROW
20 C   NUMBER IN AYZ,JDOF,EULER.
21 C       CALL READIM (IVEC, II,NJS, I,KJ)
22 C   READ XYZ.
23 C       CALL READ (BIGP, NJF,I3, KBIGP,3)
24 C       DO 5 I=1,NJS
25 C         JNT = IVEC(I)
26 C         XYT(I,1) = BIGP(JNT,1)
27 C         XYT(I,2) = BIGP(JNT,2)
28 C   READ EULER.
29 C       CALL READ (BIGP, NJF,I3, KBIGP,3)
30 C       DO 6 I=1,NJS
31 C         JNT = IVEC(I)
32 C         XYT(I,3) = BIGP(JNT,3)
33 C         CALL WRITE (XYT, NJS,3, 3HXYT, KJ)
34 C         IMS = 1
35 C         ISHIFT = 0
36 C         CALL READ (FREQM, NM,II,KM,1)
37 C   READ MODES. FIND LARGEST VALUE (ABS) IN EACH MODE. (SCALEM).
38 C       CALL YREAD (NUTP,V,LV,KV,NUTI)
39 C       CALL YSTOD (NUTP,BIGP,NBIGP,NM,KBIGP,KBIGM,V,LV,KV,NUTI)
40 C       DO 9 J=1,NM
41 C         BIGM = 0.0
42 C         DO 8 I=1,NBIGP
43 C           VALM = ABS(BIGP(I,J))
44 C           IF (VALM .GT. BIGM) BIGM=VALM
45 C         8 CONTINUE
46 C         SCALEM(J) = 1./BIGM
47 C         CALL WRITE (SCALEM, I,NM, 6HSCALEM, I)
48 C   READ JDOF.
49 C       CALL READIM (JDOF, NJF,II, KBIGP,6)
50 C   COMPRESS BIGP INTO PLOC. ORDER IS U,V IN JOINT ORDER GIVEN BY IVEC.
51 C       DO 12 I=1,NJS
52 C         JNT = IVEC(I)
53 C         IU = JDOF(JNT,1)
54 C         IV = JDOF(JNT,2)
55 C         IPU = 2*I-1
56 C         IPV = 2*I

```

```

57      DO 12 J=1,NM
58      PLOC(IPU,J) = 0.0
59      IF (IU .GT. 0) PLOC(IPU,J)=BIGP(IU,J)
60      PLOC(IPV,J) = 0.0
61      IF (IV .GT. 0) PLOC(IPV,J)=BIGP(IV,J)
62 12 CONTINUE
63      CALL WRITE (PLOC, 2*NJS,NM, 4HPLOC, KP).
64
C      C DEFINE UNDISTURBED SHAPE. XP,YP IN PLOT X,Y. LEFT,RIGHT.
65      NJ = NJS
66      NJM1 = NJ-1
67      IJ = NJ+1
68      DO 10 IP=1,NJM1
69      IJ = IJ-1
70      XP(IP) = -XYT(IJ,2)
71      10 YP(IP) = XYT(IJ,1)
72      DO 15 IJ=1,NJ
73      IP = NJM1+IJ
74      XP(IP) = XYT(IJ,2)
75      15 YP(IP) = XYT(IJ,1)
76
C      DO 199 IM=1,NM
77      MODENO = IMS-1+IM
78      SCALE = SCALEM(IM)
79      FREQ = FREQM(IM)
80
C      CALCULATE DELTA-X, DELTA-Y. IN JAG X,Y. LEFT=DXL,DYL. RIGHT.
81      CALL PAGEHD
82      WRITE (NOT,2002) MODENO
83      DO 50 IJ=1,NJ
84      IXL = 2*IJ-1
85      IYL = 2*IJ
86      ANG = XYT(IJ,3)/57.29577951
87      CJ = COS(ANG)
88      SJ = SIN(ANG)
89      DXJAG = PLOC(IXL,IM)*CJ - PLOC(IYL,IM)*SJ
90      DYJAG = PLOC(IXL,IM)*SJ + PLOC(IYL,IM)*CJ
91      WRITE (NOT,2001) DYJAG,DXJAG
92      DX(IJ) = XYT(IJ,1) + SCALE*DXJAG
93      DXL(IJ) = XYT(IJ,1) - SCALE*DXJAG
94      DY(IJ) = XYT(IJ,2) + SCALE*DYJAG
95      50 DYL(IJ) = XYT(IJ,2) + SCALE*DYJAG
96
C      PACK DY,DX INTO XP,YP. LEFT,RIGHT.
97      IP1 = NJM1+NJ+1
98      IP2 = IP1-1+NJM1
99      IJ = NJ+1
100     DO 60 IP=IP1,IP2
101     IJ = IJ-1
102     XP(IP) = DYL(IJ)
103     60 YP(IP) = DXL(IJ)
104     IP1 = NJM1+NJ+NJM1
105     DO 65 IJ=1,NJ
106     IP = IP1+IJ
107     XP(IP) = DY(IJ)
108     65 YP(IP) = DX(IJ)
109
C      PACK XP,YP FOR PRINTING.
110     NJ2M1 = 2*NJ-1
111     DO 80 I=1,NJ2M1
112

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
114      XPPY(I,1) = XP(I)
115      XPPY(I,2) = YP(I)
116      XPPY(I,3) = 0.0
117      XPPY(I,4) = XP(I+NJ2M1)
118 80   XPPY(I,5) = YP(I+NJ2M1)
119      CALL WRITE(XPPY, NJ2M1, 5, 4HXYP, KP)
120      WRITE(NOT,2003) MODENO,SCALE
121      C PUT IN TANK TOP,BOT AT CENTERLINE.
122      XP(IP+1) = 0.
123      YP(IP+1) = 16.
124      XP(IP+2) = 0.
125      YP(IP+2) = 0.
126      NRP = IP+2
127      IFSAME = 1
128      IFCURV = 1
129      IFLIFT = 1
130      ENCODE(40,5001,PTITLE) MODENO,FREQ,ISHIFT,SCALE
131      CALL PLOT1(XP,YP,NRP,1, -10.,2., 6HY-TANK,6HX-TANK,PTITLE,
132      *           IFSAME,IFCURV,IFLIFT,KPLOT)
133      199 CONTINUE
134      C
135      CALL PLTND
136      RETURN
137      END
```

@PRT F1.0YMODE

```

SIRISHP00000*F1•OYMODE
1      SUBROUTINE OYMODE
2
3      C MAIN PROGRAM TO TEST ITERATIVE RAYLEIGH-RITZ METHOD OF DR. JOHN ADMIRE
4      C TECHNIQUE = COMPOSITE STRUCTURE.
5      C VERSION = NON-SWEEPING.
6      C PROGRAMMING LOGIC = SPARSE.
7      C MAXIMUM SIZE OF MASS,STIF = 1820.
8      C MAXIMUM NU = 70
9      C DEVELOPED BY R L WOHLEN AND R A PHILIPPUS. MARCH 1972.
10     C LAST DIMENSION CHANGE (FOR 150000) BY HRUDA 02APR74.
11     C LAST REVISION BY R A PHILIPPUS. JULY 1974.
12
13     C *****
14     C INPUT DATA READ IN THIS PROGRAM.
15     C          NW           FORMAT (10X,15)
16     C          NU           FORMAT (10X,15)
17     C          SHIFT        FORMAT (10X,E10)
18     C          MAXIT       FORMAT (10X,15)
19     C          IPUNCH      FORMAT (A6)
20     C          RETURN
21
22     C DEFINITION OF INPUT VARIABLES.
23     C NW      = NUMBER OF MODES WANTED.
24     C NU      = NUMBER OF RAYLEIGH-RITZ MODES TO USE.
25     C SHIFT   = SHIFT VALUE TO USE.
26     C MAXIT   = MAXIMUM NUMBER OF ITERATIONS TO BE PERFORMED.
27     C IPUNCH  = PUNCH CARD OPTION FOR W2, FREQ, MODES, AND TMODES.
28     C          = 6HPUNCH, FOR PUNCH CARD OUTPUT.
29     C          = 6HNOPUNC, NO PUNCH CARD OUTPUT.
30
31     C          COMMON /BTAPEM/ NUTM,NUTK,NUTTR,NUTZ,NUTF,NUTI,NUT2,NUT3,NUT4,
32     *          NUT5,NUT6,NUT7
33     C          COMMON / RESTAP / NR5VT1
34     C          COMMON / RTRANS / IFTRAN
35
36     C          DIMENSION V(10920), LV(10920), W(70), FREQ(70),
37     *          AL(70, 70), S( 70, 70), MH(10)
38
39     C          EQUIVALENCE (V(3641),S), (LV(3641),AL)
40
41     C          DATA NIT,NOT / 5,6 /
42     C          DATA KV, KA /
43     *          10920, 70 /
44     C          DATA NITER1, NITER2, TULZ, TULWZ/
45     *          0,      1, 1.E-06, 1.E-04/
46     C          DATA IFPRNT/1000/
47     1001 FORMAT (10X, 4I5)
48     1010 FORMAT (10X, E10.0)
49     1020 FORMAT (12A6)
50
51     C          READ (NIT,1001) NW
52     C          READ (NIT,1001) NU
53     C          READ (NIT,1010) SHIFT
54     C          READ (NIT,1001) MAXIT
55     C          READ (NIT,1020) IPUNCH
56

```

```

57      CALL YM0DEZ (NUTM,NUTK,NUT2,W2,W,FREQ,NW,V,LV,A,S,KV,KA,
58      *          NUT1,NUT2,NUT3,NUT4,NUT5,NUT6,NUT7,
59      *          IFFRNT,MAXIT,
60      *          NU,NITER1,NITER2,SHIFT,TOLZ,TOLW2)
61
62      C
63      CALL WRITE (W2,NW,1,ZHW2,KA)
64      CALL WRITE (W,NW,1,1HW,KA)
65      CALL WRITE (FREQ,NW,1,4HFREQ,KA)
66      CALL YWRITE (NUTZ,5HM0DES,V,LV,KV)
67      IF (NRSVT1 .GT. 0) CALL WTape (W2,NW,1,ZHW2,KA,NRSVT1)
68      IF (NRSVT1 .GT. 0) CALL WTape (FREQ,NW,1,4HFREQ,KA,NRSVT1)
69      IF (NRSVT1 .GT. 0) CALL YTape (NUTZ,5HM0DES,V,LV,KV,NRSVT1)
70      CALL YT0S (FREQ,NUTF,NW,1,KA,1,V,LV,KV,NUT7)
71      IF (IFTRAN .EQ. 0) GO TO 100
72      REWIND NUTR
73      CALL YINI (NUTTR,MH,1,1D)
74      NRT = MH(1)
75      NCT = MH(2)
76      CALL YD1SA (NUTZ,1,1,NUT1,NCT,NW,V,LV,KV,NUT7)
77      CALL YMUL T (NUTTR,NUT1,NUT2,V,LV,KV,NUT7)
78      REWIND NUTZ
79      CALL YINI (NUTZ,NRM,1,1)
80      NRTM = NRM - NCT + NRT
81      CALL YZERO (NUT1,NRTM,NW)
82      CALL YASSEM (NUTZ,1,1,NUT1,V,LV,KV,NUT5,NUT6,NUT7)
83      NCTP1 = NCT + 1
84      NRX = NRM - NCT
85      CALL YD1SA (NUTZ,NCTP1,1,NUT3,NRX,NW,V,LV,KV,NUT7)
86      NRTP1 = NRT + 1
87      CALL YASSEM (NUT3,NRTP1,1,NUT1,V,LV,KV,NUT5,NUT6,NUT7)
88      CALL YWRITE (NUT1,6HTM0DES,V,LV,KV)
89      IF (NRSVT1 .GT. 0) CALL YTape (NUT1,6HTM0DES,V,LV,KV)
90      100 IF (NRSVT1 .GT. 0) CALL LTape (NRSVT1)
91      IF (IPUNCH .NE. 5HPUNCH) RETURN
92      CALL PUNCH (FREQ,NW,1,4HFREQ,KA)
93      CALL YPUNCH (NUTZ,5HM0DES,V,LV,KV)
94      IF (IFTRAN .NE. 0) CALL YPUNCH (NUT1,6HTM0DES,V,LV,KV)
95      RETURN
96      C
97      END

```

```
SIRISHP00000*F1•PBASIC
1      SUBROUTINE PBASIC
2      C PROGRAM TO CALL BASIC,
3      COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
4      DIMENSION XYZ(2000,3), JDOF(2000,6), EUL(2000,3)
5      DATA KRX, KCX, KRJ, KCJ, KRE, KCE /
6      *    2000,   3, 2000,   6, 2000,   3 /
7      C
8      CALL BASIC  (XYZ,JDOF,EUL,KRX,KCX,KRJ,KCJ,KRE,KCE,
9      *           NUTEL,NUTXYZ)
10     RETURN
11     END
```

WPRTR F1•PXTRA

```

SIRISHP00000*F1•PATRA
1      SUBROUTINE PXTTRA
2      C  OVERLAY PROGRAM TO REVADD EXTRA MASS AND STIFFNESS MATRICES TO
3      C  EXISTING MASS AND STIFFNESS MATRICES.
4      C
5      C ***** INPUT DATA READ IN THIS PROGRAM. *****
6      C
7      C      CALL READIM (IJVEC,I,NCI)
8      C      CALL YREAD (MASS MATRIX)
9      C      CALL YREAD (STIF MATRIX)
10     C      RETURN
11     C
12     COMMON / BTAPET / NUTM,NUTK,NUT1,NUT2,NUT3,NUT4,NUTX
13     DIMENSION V(12000), LV(12000), IJVEC(2000)
14     DATA KV, KIV /
15     *    12000, 2000 /
16     C  READ IJVEC FOR EXTRA M,K MATRICES.
17     C      CALL READIM (IJVEC,NRI,NCI,I,KIV)
18     DO 10 I=1,NCI
19     IF (IJVEC(I),NE. 0) GO TO 20
20     C  CONTINUE
21     DO 15 I=1,NCI
22     15 IJVEC(I) = I
23     C  READ EXTRA MASS MATRIX.
24     20 CALL YREAD (NUTX,V,LV,KV,NUT1)
25     C      CALL YREVAD (1*,NUTX,IJVEC,IJVEC,NUTM,V,LV,KV,NUT1,NUT2,NUT3,NUT4)
26     C  READ EXTRA STIFFNESS MATRIX.
27     C      CALL YREAD (NUTX,V,LV,KV,NUT1)
28     C      CALL YREVAD (1*,NUTX,IJVEC,IJVEC,NUTK,V,LV,KV,NUT1,NUT2,NUT3,NUT4)
29     C      CALL YWRITE (NUTM,5HMXTRA ,V,LV,KV)
30     C      CALL YWRITE (NUTK,5HKXTRA ,V,LV,KV)
31     C      RETURN
32     END

```

WPRTR F1•REDUCE

SIRISHP00000*F1.REDUCE

SUBROUTINE REDUCE

```

1      C
2      C OVERLAY PROGRAM TO REDUCE STIFFNESS AND MASS MATRIX.
3      C DOF TO BE REDUCED MUST BE POSITIONED FIRST IN MATRIX.
4      C REDUCING TRANSFORMATION IS STORED ON NUTTR.
5      C DEVELOPED BY WA BENFIELD, FEBRUARY 1974.
6      C
7      C *****
8      C INPUT DATA READ IN THIS PROGRAM.
9      C
10     C NR                                     FORMAT (10X,15)
11     C CALL READIM (IJVEC,1,NCI)
12     C RETURN
13     C
14     C DEFINITION OF INPUT VARIABLES.
15     C NR      = NUMBER OF ROW-COLS IN REDUCED MATRIX.
16     C IJVEC   = IJVEC TO REARRANGE ROWS AND COLS BEFORE REDUCING.
17     C
18     C COMMON /BTAPER / NUTMR,NUTKR,NUTTR,NUTR1,NUTR2,NUTR3,NUTR4,NUTR5,
19     *          NUTR6
20     C COMMON /RTRANS / IFTRAN
21     C
22     C DIMENSION V(10000), LV(10000)
23     C DIMENSION IV1(3000), IV2(3000)
24     C
25     C DATA NIT, NOT / 5, 6 /
26     C DATA KV / 10000 /
27     C DATA KIV / 3000 /
28     C
29     C 1001 FORMAT (6(10X,15))
30     C 2001 FORMAT (/// 20X14)OVERLAY REDUCE,
31     *           // 10X42HNUMBER OF ROWS AND COLS BEFORE REDUCING = ,15,
32     *           / 10X42HNUMBER OF ROWS AND COLS AFTER REDUCING = ,15)
33     *           DO 10 I=1,KIV
34     10 IV2(I) = I
35     C
36     C READ NUMBER OF ROWS IN REDUCED MATRIX.
37     C READ (NIT,1001) NR
38     C REWIND NUTKR
39     C CALL YINI (NUTKR,NRK,1,1)
40     C CALL PAGEHD
41     C WRITE (NOT+2001) NRK, NR
42     C
43     C READ REARRANGING IVEC.
44     C CALL READIM (IV1,NRI,NCI,1,KIV)
45     C DO 12 I=1,NCI
46     C   IF (IV1(I) .NE. 0) GO TO 17
47     C 12 CONTINUE
48     C   DO 15 I=1,NCI
49     C     15 IV1(I) = I
50     C
51     C REARRANGE MASS, STIFF, AND TRANS (IF ANY) MATRICES.
52     C 17 CALL YZERO (NUTR1,NRK,NRK)
53     C   CALL YZERO (NUTR2,NRK,NRK)
54     C   CALL YREVAD (1.0,NUTMR,IV1,IV1,NUTR1,V,LV,KV,NUTR2,NUTR3,NUTR4,
55     *                  NUTR5)
56     C   CALL YREVAD (1.0,NUTKR,IV1,IV1,NUTR2,V,LV,KV,NUTR3,NUTR4,NUTR5,

```

```
57      * NUTMR)
58      IF (IFTRAN .EQ. 0) GO TO 20
59      REWIND NUTTR
60      CALL YINI (NUTTR, LV, 1, 2)
61      CALL YZERO (NUTR3, LV(1)), LV(2))
62      CALL YREVAD (1.0, NUTTR, IV2, IV1, NUTR3, V, LV, KV, NUTR4, NUTR5, NUTMR,
63      * NUTKR)
64      20 CALL YSHEDZ (NUTR2, NUTKR, NUTTR, NR, 1, V, LV, KV, NUTR4, NUTR5, NUTR6,
65      * NUTMR)
66      CALL YWRITE (NUTTR, 2HTR , V, LV, KV)
67      CALL YWRITE (NUTKR, 2HKR , V, LV, KV)
68      IF (IFTRAN .EQ. 0) GO TO 30
69      CALL YMULTB (NUTR3, NUTTR, V, LV, KV, NUTR4, NUTR5)
70      CALL YWRITE (NUTTR, 6HMULTTR, V, LV, KV)
71      30 CALL YTAB (NUTR1, NUTTR, NUTMR, V, LV, KV, NUTR4, NUTR5)
72      CALL YWRITE (NUTMR, 2HMR , V, LV, KV)
73      IFTRAN = 1
74      RETURN
75      END
```

BPRT F1.SAVE

```

STRISHPO00000*F1.SAVE
1      SUBROUTINE SAVE
2
3      C OVERLAY PROGRAM TO SAVE MASS, STIFFNESS, AND REDUCING TRANSFORMATION
4      C MATRICES ON FORMA LIBRARY TAPE.
5      C DEVELOPED BY WA BENFIELD, FEBRUARY 1974.
6
7      C *****
8      C INPUT DATA READ IN THIS PROGRAM.
9      C      NAMEM,NAMEK,NAMET                                FORMAT (3A6)
10     C      RETURN
11     C
12     C DEFINITION OF INPUT VARIABLES.
13     C      NAMEM = NAME OF MASS MATRIX TO USE ON FORMA TAPE.
14     C      = 6H      , MASS IS NOT WRITTEN ON FORMA TAPE.
15     C      NAMEK = NAME OF STIF MATRIX TO USE ON FORMA TAPE.
16     C      = 6H      , STIF IS NOT WRITTEN ON FORMA TAPE.
17     C      NAMET = NAME OF TRANSFORMATION MATRIX TO USE ON FORMA TAPE.
18     C      = 6H      , TRANSFORMATION IS NOT WRITTEN ON FORMA TAPE.
19     C
20     C      COMMON / BTAPES / NUTMS,NUTKS,NUTTS
21     C      COMMON / RESTAP / NRSVT1
22     C      COMMON / RTRANS / IFTRAN
23     C
24     C      DIMENSION V(15000), LV(15000)
25     C
26     C      DATA KV / 15000 /
27     C      DATA NIT, NOT / 5, 6 /
28
29     C      1001 FORMAT (12A6).
30
31     C      READ (INIT,1001) NAMEM,NAMEK,NAMET
32     C      IF (NAMEM .NE. 6H)      CALL YWTAPE (NUTMS,NAMEM,V,LV,KV,NRSVT1)
33     C      IF (NAMEK .NE. 6H)      CALL YWTAPE (NUTKS,NAMEK,V,LV,KV,NRSVT1)
34     C      IF (IFTRAN .NE. 0 .AND. NAMET .NE. 6H)      CALL YWTAPE (NUTTS,
35                                         NAMET,V,LV,KV,NRSVT1)
36
37     C      CALL LTAPE (NRSVT1)
38     C      RETURN
39     C      END

```

MPRT F1-SUBSTR

```

SIRISHPOOOOO*F1*SUBSTR
1      SUBROUTINE SUBSTR
2
3      C OVERLAY PROGRAM TO COMBINE SUBSTRUCTURE MASS AND STIFFNESS MATRICES
4      C TOGETHER.
5      C DEVELOPED BY WA BENFIELD. FEBRUARY 1974.
6
7      C ****=  

8      C INPUT DATA READ IN THIS PROGRAM.
9      C      NSUBS                                FORMAT (10X,15)
10     C      NR                                  FORMAT (1DX,15)
11     C      NTRAN                               FORMAT (A6)
12     C      DO 30 K=1,NSUBS
13     C      NSRC                                FORMAT (10X,15)
14     C      CALL YREAD (MASS MATRIX)
15     C      30 CALL YREAD (STIF MATRIX)
16     C      IF (NTRAN .EQ. 6HTRANS ) CALL YREAD (TRANS MATRIX)
17     C      RETURN
18
19     C DEFINITION OF INPUT VARIABLES.
20     C      NSUBS = NUMBER OF SUBSTRUCTURES TO BE READ IN.
21     C      NR    = NUMBER OF ROW-COLS IN TOTAL MASS-STIF MATRICES.
22     C      NTRAN = OPTION TO READ IN TRANSFORMATION MATRIX,
23     C              = 6HTRANS , TRANSFORMATION MATRIX IS READ IN.
24     C              = 6HNOTRAN, TRANSFORMATION MATRIX IS NOT READ IN.
25     C      NSRC   = START ROW-COL TO ASSEMBLE SUBSTRUCTURE INTO.
26
27     COMMON / BTAPEP / NUTMP,NUTKP,NUTTP,NUT1P,NUT2P,NUT3P,NUT4P,NUT5P
28     COMMON / RTRANS / ITRANS
29
30     DIMENSION V(10000), LV(10000)
31     DIMENSION IV1(3000)
32
33     DATA NIT, NOT / 5, 6 /
34     DATA KV / 10000 /
35     DATA KIV / 3000 /
36
37     1001 FORMAT (6(10X,15))
38     1002 FORMAT (12A6)
39
40     C READ NUMBER OF SUBSTRUCTURES TO COMBINE TOGETHER.
41     READ (NIT,1001) NSUBS
42
43     C READ NUMBER OF ROW-COLS IN FINAL MATRIX, AND IF TRANSFORMATION WILL
44     C BE INPUT.
45     READ (NIT,1001) NR
46     READ (NIT,1002) NTRAN
47     CALL YZERO (NUTMP,NR,NR)
48     CALL YZERO (NUTKP,NR,NR)
49     DO 30 K=1,NSUBS
50
51     C READ STARTING ROW-COL NUMBER TO ASSEMBLE SUBSTRUCTURE INTO.
52     READ (NIT,1001) NSRC
53     IF (NSUBS .EQ. 1 .AND. NSRC .EQ. 1) GO TO 40
54     DO 10 I=1,NR
55     10 IV1(I) = NSRC + I - 1
56

```

```
57      C READ SUBSTRUCTURE MASS MATRIX.  
58      CALL YREAD (NUT1P,V,LV,KV,NUT2P)  
59      CALL YREVAD (1.0,NUT1P,IV1,IV1,NUTMP,V,LV,KV,NUT2P,NUT3P,NUT4P,  
60          NUT5P)  
61      C  
62      C READ SUBSTRUCTURE STIFF MATRIX.  
63      CALL YREAD (NUT1P,V,LV,KV,NUT2P)  
64      CALL YREVAD (1.0,NUT1P,IV1,IV1,NUTKP,V,LV,KV,NUT2P,NUT3P,NUT4P,  
65          NUT5P)  
66      30 CONTINUE  
67      CCCCC CALL YWRITE (NUTKP,6HTOTSTF,V,LV,KV)  
68      CCCCC CALL YWRITE (NUTMP,6HTOTMAS,V,LV,KV)  
69      GO TO 50  
70      40 CALL YREAD (NUTMP,V,LV,KV,NUT2P)  
71      CALL YREAD (NUTKP,V,LV,KV,NUT2P)  
72      C  
73      C READ TRANSFORMATION MATRIX (IF ANY).  
74      50 IF (INTRAN .NE. 6HTRANS) RETURN  
75      CALL YREAD (NUTTP,V,LV,KV,NUT2P)  
76      IFTRAN = 1  
77      RETURN  
78      END
```

OPRT F1•XYZEU2

```

SIRISHP00000*F1*XYZEU2
1      SUBROUTINE XYZEU2
2      C
3      C      SUBROUTINE XYZJAG (NUTEL,NUTXYZ)
4      C
5      DIMENSION XYZ(1200,3),EUL(1200,3),JDOOF(1200,6)
6      DIMENSION ICONF(1500,8),ICONS(200,4),ICONB(200,4)
7      DIMENSION XYZB(200,3),EULB(200,3)
8      DIMENSION RB(20),AI(20),X0(20),NP INTS(20),NELEM(20),X0I(20)
9      DIMENSION ANS(10),AN6$11,INTG(2)

10     C
11     COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
12     C
13     DATA KX,KS,KB/1200,200,20/
14     DATA PENTA,HEXA/5HPENTA,4MHEXA/
15     DATA NCX,NCJ/3,6/
16     DATA AN6/6HRETURN,6HFLUID ,6HGRAVITY,6H      ,6HTRNGL ,6HQUAD +
17     *      6HMI   ,6HMZ   ,6HK1   ,6HK2   ,6HCNST   /,
18     *      AN5/5HRO   ,5HBLKM ,5HGVX   ,5HGVY   ,5HGVZ   ,5HE    ,5HNU   ,
19     *      SHTMAS ,SHTMEM ,SHTBEN   /,
20     *      INTG/0,0/
21     DATA ZZERO/0.0/
22     DATA EPS/1.0E-05/
23     C
24     C
25     1001 FORMAT (16I5)
26     1002 FORMAT (8F10.0)
27     1003 FORMAT (8A10)
28     3001 FORMAT (A6)
29     3002 FORMAT (5(A6,4X))
30     3003 FORMAT (3(A5,E10.3))
31     3004 FORMAT (4I5,3E10.3)
32     3005 FORMAT (5I5,3E10.3)
33     3006 FORMAT (9I5)
34     5001 FORMAT (10I/)
35     *      15X, 22H RADIUS OF THE SPHERE =, F10.3,//
36     *      15X, 22H FLUID MASS DENSITY =, F10.7,//
37     *      15X, 22H      BULK MODULUS =, F14.3,//
38     *      15X, 22H          GVX =, F10.3,//
39     *      15X, 22H          GVV =, F10.3,//
40     *      15X, 22H          GVZ =, F10.3,//
41     *      15X, 22H BLADDER MASS DENSITY =, F10.7,//
42     *      15X, 22H          E =, F14.3,//
43     *      15X, 22H          NU =, F10.7,//
44     *      15X, 22H BLADDER THICKNESS =, F10.7)
45     5002 FORMAT (10I/)
46     *      15X, 22H          NCIR =, 15,//
47     *      15X, 22H          NRBAY =, 15)

48     C
49     C -----
50     C
51     C      I N P U T S -----
52     C
53     C      READ (5,1002) R                      (8F10.0)
54     C      READ (5,1002) RHOF,BLKM,GVX,GVV,GVZ   (8F10.0)
55     C      READ (5,1002) RHOBLD,EBLD,ANUBLD,TBLD   (8F10.0)
56     C      READ (5,1001) NCIR,NRBAY                (16I5)

```

```

57      C CALL READ (AI,NRA,NCA,1,KB)          (5 OF THEM)
58      C CALL READ (RB,NRR,NCR,1,KB)          (1 X NRBAY)
59      C CALL READIM (NPINTS, NRN,NCN,1,KB)    (1 X (NRBAY+1))
60      C CALL READIM (NELEM1,NRE,NCE,1,KB)    (1 X NRBAY)
61      C READ (5,1002) (XO(I), I=1,NP1)        (8F10.0)
62      C READ (5,1002) (XOI(I),I=1,NP2)        (8F10.0)
63      C DO 200 II=2,NRBAY
64      C READ (5,1002) (XOI(K),K=1,NP1)        (8F10.0)
65      C 200 CONTINUE
66      C READ (5,3002) ELMID1,ELMID2          (5(A6,4X))
67      C READ (5,3002) ELMID1,ELMID2          (5(A6,4X))
68      C DO 500 II=3,NRBAY1
69      C READ (5,3002) ELMID1,ELMID2          (5(A6,4X))
70      C 500 CONTINUE
71      C -----
72      C -----
73      C -----
74      C EXPLANATIONS -----
75      C-
76      C- AI=COEFFICIENTS OF THE FREE SURFACE POLYNOMIAL.
77      C- R=RADIUS OF THE SPHERE.
78      C- NCIR=NO. OF CIRCUMFERENTIAL PTS. AROUND THE MODEL.
79      C- NRBAY=NO. RADIAL BAYS IN THE MODEL.
80      C- RB=RADIAL BAY RADII
81      C- THETAT=TOTAL MODEL ANGLE (=180.0)
82      C- NPOINTS=NO. OF POINTS ON EACH VERTICAL LINE IN THE PLANE. (VECTOR)
83      C- XO=XO-S FOR THE CENTERLINE.
84      C- XOI=X-VALUES IN EACH VERTICAL LINE IN THE PLANE (NO.=NPOINTS)
85      C- (NOTE: POINTS ON SPHERE AND BLADDER ARE REPLACED BY ACTUAL
86      C- CALCULATED VALUES). STARTING FROM IN EXCEPTING CENT.LINE.
87      C- ELMID1=FIRST ELEMENT (FROM BOTTOM) FOR THE ANNULAR SPACE ID.
88      C- ELMID2=LAST ELEMENT (FROM BOTTOM) FOR THE ANNULAR SPACE ID.
89      C- PENTA=PENTAHEDRON
90      C- HEXA=HEXAHEDRON
91      C- NELEM1=NO. OF ELEMENTS IN EACH BAY (PLANER MODEL) (VECTOR)
92      C-
93      C-
94      C- BLADDER IS SUPPOSED TO BE ATTACHED AT DIAMETRAL PLANE
95      C-
96      C-
97      C- RHOFL=MASS DENSITY OF THE FLUID
98      C- BLKFL=BULK MODULUS FOR THE FLUID
99      C- GVX=GRAVITY ACCELERATION IN X-DIRECTION
100     C- GYV=GRAVITY ACCELERATION IN Y-DIRECTION
101     C- GVZ=GRAVITY ACCELERATION IN Z-DIRECTION
102     C- RHOBLD=MASS DENSITY OF THE BLADDER
103     C- EBLD=YOUNG'S MODULUS FOR THE BLADDER.
104     C- ANUBLD=POISSON'S RATIO FOR THE BLADDER
105     C- TBLD=THICKNESS OF THE BLADDER
106     C-
107     C- NOTE=THE TOP ELEMENTS IN THE FIRST ANNULAR SPACE HAVE TO BE TETRAS.
108     C-
109     C-----
110     C-
111     C- REWIND NUTEL
112     C- REWIND NUTXYZ
113     C-

```

```

114      PI = ATAN2(0.0,-1.0)
115      RADDEG = ATAN2(1.0,1.0) / 45.0
116      C-
117      C-  INITIALIZE SOME VARIABLES REQUIRED
118      C
119      NDP = 0
120      ND = 0
121      NEL = 0
122      NELS= 0
123      NELB= 0
124      NDPB= 0
125      C
126      CALL ZERO (XYZ,KX,3,KX)
127      CALL ZERO (XYZB,KS,3,KS)
128      CALL ZERO (EUL,KX,3,KX)
129      CALL ZERO (EULB,KS,3,KS)
130      DO 20 J=1,1500
131      DO 20 J=1,8
132      20 ICONF(I,J) = 0
133      DO 30 I=1,KS
134      DO 30 J=1,8
135      ICONS(I,J) = 0
136      30 ICONB(I,J) = 0
137      C
138      C-  READ IN DATA
139      C
140      READ (5,1002) R
141      READ (5,1002) RHOF,BLKM,GVX,GVY,GVZ
142      READ (5,1002) RHOBLD,EBLD,ANUBLD,TBLD
143      C
144      READ (5,1001) NCIR,NRBAY
145      C
146      WRITE (6,5001) R,RHOF,BLKM,GVX,GVY,GVZ,RHOBLD,EBLD,ANUBLD,TBLD
147      WRITE (6,5002) NCIR,NRBAY
148      C
149      CALL READ (AI,NRA,NCA,I,KB)
150      CALL READ (RB,NRR,NCR,I,KB)
151      CALL READIM (NPINTS, NRN,NCN,I,KB)
152      CALL READIM (NELEMI,NRE,NCE,I,KB)
153      C
154      NCIRI = NCIR + 1
155      ANCIR = NCIR
156      ANCIRI = NCIRI
157      THETAT = PI
158      THETAD = THETAT / ANCIRI
159      NDV2 = NCIRI
160      NQ = NCIRI / 2
161      NQ1 = NQ - 1
162      NQ2 = NQ + 1
163      C
164      C  TOTAL NO. OF FLUID POINTS.
165      NTPF = 0
166      DO 15 K=2,NCN
167      15 NTPF = NTPF + NPINTS(K)
168      NTPF = (NTPF*NCIR) + NPINTS(1)
169      C-  POINTS IN THE FIRST ANNULAR SPACE
170      NP1 = NPINTS(1)

```

```

171      NP2 = NPINTS(2)
172      READ (5,1002) (X0(I), I=1, NP1)
173      READ (5,1002) (X0I(I), I=1, NP2)
174
175      C
176      Y1 = 0.0
177      X81 = R - SQRT(R**2-Y1**2)
178      W1 = 0.0
179      W1 = AI(1) + AI(2)*Y1 + AI(3)*Y1**2 + AI(4)*Y1**3 + AI(5)*Y1**4
180      XT1 = R + SQRT(R**2-Y1**2) - W1
181
182      C
183      Y2 = RB(1)
184      X82 = R - SQRT(R**2-Y2**2)
185      W2 = 0.0
186      W2 = AI(1) + AI(2)*Y2 + AI(3)*Y2**2 + AI(4)*Y2**3 + AI(5)*Y2**4
187      XT2 = R + SQRT(R**2-Y2**2) - W2
188
189      C
190      X0(1) = X81
191      X0(NP1) = XT1
192      X0I(1) = X82
193      X0I(NP2) = XT2
194
195      NDPB = 1
196      C- FIRST POINT OF THE MODEL
197      ND = ND + 1
198      JDOOF(1,1) = ND
199      EUL(1,3) = (PI/2.0) / RADDEG
200      C- ALL OTHERS IN THE FIRST BAY.
201      NDP = 1
202      DO 60 I=1,NP2
203      THETA = 0.0
204      DO 80 J=1,NCIR
205      NDP = NDP + 1
206      XYZ(NDP,1) = X0I(I)
207      XYZ(NDP,2) = RB(I) * COS(THETA)
208      XYZ(NDP,3) = RB(I) * SIN(THETA)
209      IF (I.NE.NPINTS(2)) GO TO 65
210      EUL(NDP,1) = ATAN2(XYZ(NDP,3),XYZ(NDP,2)) / RADDEG
211      EUL(NDP,2) = 0.0
212      IF (J .GT. 1) GO TO 82
213      TERM = (R**2 - XYZ(NDP,2)**2)
214      * IF (ABS(TERM) .LT. EPS) DEG = -90.0
215      * IF (ABS(TERM) .LT. EPS) GO TO 82
216      * TERM1 = (1.0/2.0) * (R**2 - XYZ(NDP,2)**2) ** (-0.5) *
217      *           (-2.0*XYZ(NDP,2))
218      *           - (AI(2) + 2.0*AI(3)*XYZ(NDP,2) + 3.0*AI(4)*XYZ(NDP,2)**2
219      *           + 4.0*AI(5)*XYZ(NDP,2)**3)
220      * DEG = ATAN(TERM1) / RADDEG
221      82 CONTINUE
222      EUL(NDP,3) = -90.0 - DEG
223
224      C
225      NDPB = NDPB + 1
226      XYZB(NDPB,1) = XYZ(NDP,1)
227      XYZB(NDPB,2) = XYZ(NDP,2)
228      XYZB(NDPB,3) = XYZ(NDP,3)
229      EULB(NDPB,1) = EUL(NDP,1)
230      EULB(NDPB,2) = EUL(NDP,2)
231      EULB(NDPB,3) = EUL(NDP,3)

```

ORIGINAL PAGE
OF POOR QUALITY

```

228
229      65 CONTINUE
230      IF (I .NE. 1) GO TO 66
231      EUL(NDP,1) = ATAN2(XYZ(NDP,3),XYZ(NDP,2)) / RADDEG
232      EUL(NDP,3) = -ASIN((2+0*(XYZ(NDP,1)-R)) /
233      1          (4.0*(XYZ(NDP,1)-R)**2 + 4.0*XYZ(NDP,3)**2 +
234      2          4.0*XYZ(NDP,2)**2)**0.5) / RADDEG
235      IF (J .GT. NQ2) GO TO 79
236      IF (J.EQ.1 .OR. J.EQ.NQ2) GO TO 76
237      ND = ND + 1
238      JDOOF(NDP,1) = ND
239      ND = ND + 1
240      JDOOF(NDP,3) = ND
241      GO TO 79
242      76 CONTINUE
243      IF (J .EQ. 1) KD = 1
244      IF (J .EQ. NQ2) KD = 3
245      ND = ND + 1
246      JDOOF(NDP,KD) = ND
247      GO TO 79
248      66 CONTINUE
249      IF (J .GT. NQ2) GO TO 79
250      IF (J.EQ.1 .OR. J.EQ.NQ2) GO TO 75
251      ND = ND + 1
252      JDOOF(NDP,1) = ND
253      ND = ND + 1
254      JDOOF(NDP,2) = ND
255      ND = ND + 1
256      JDOOF(NDP,3) = ND
257      GO TO 79
258      75 CONTINUE
259      IF (J .NE. 1) GO TO 77
260      ND = ND + 1
261      JDOOF(NDP,1) = ND
262      ND = ND + 1
263      JDOOF(NDP,2) = ND
264      GO TO 79
265      77 CONTINUE
266      IF (I .NE. NP2) KD = 2
267      IF (I .EQ. NP2) KD = 3
268      ND = ND + 1
269      JDOOF(NDP,KD) = ND
270      79 CONTINUE
271      THETA = THETA + THETAD
272      80 CONTINUE
273      IF (I .EQ. 1 .OR. I .EQ. NP2) GO TO 1082
274      NDP = NDP - NCIR
275      N1 = NDP
276      C
277      N2 = NDP + NDV2 + 2
278
279      N1 = N1 + 1
280      N2 = N2 - 1
281      C
282      JDOOF(N2,1) = -JDOOF(N1,1)
283      JDOOF(N2,2) = JDOOF(N1,2)
284      DO B1 J = 1,NQ1
285      N1 = N1 + 1
286      N2 = N2 - 1

```

```

285      C
286          JDOF(N2,1) = -JDOF(N1,1)
287          JDOF(N2,2) = JDOF(N1,2)
288          JDOF(N2,3) = -JDOF(N1,3)
289      81 CONTINUE
290      GO TO 1084
291 1082 CONTINUE
292      C
293          NDP = NDP + NCIR
294          N1 = NDP
295          N2 = NDP + NDV2 + 2
296      C
297          N1 = N1 + 1
298          N2 = N2 - 1
299          JDOF(N2,1) = -JDOF(N1,1)
300          JDOF(N2,2) = -JDOF(N1,2)
301      C
302      DO 1083 J = 1,NQ1
303          N1 = N1 + 1
304          N2 = N2 - 1
305      C
306          JDOF(N2,1) = -JDOF(N1,1)
307          JDOF(N2,2) = -JDOF(N1,2)
308          JDOF(N2,3) = JDOF(N1,3)
309 1083 CONTINUE
310      C
311 1084 CONTINUE
312      C
313          NDP = NDP + NCIR
314      C- CENTERLINE POINTS
315          NDP = NDP + 1
316          XYZ(NDP,1) = X0(I+1)
317          ND = ND + 1
318          JDOF(NDP,2) = ND
319      60 CONTINUE
320      C-- FOR THE TOP POINT (TOUCHING BLADDER)
321          JDOF(NDP,1) = JDOF(NDP,2)
322          JDOF(NDP,2) = 0
323          XYZB(1,1) = XYZ(NDP,1)
324          EULB(1,3) = (-PI/2,0) / RADDEG
325      C
326          EUL(NDP,3) = EULB(1,3)
327      C- ALL OTHER NODE POINTS STARTING FROM SECOND BAY OUTWARDS.
328
329      DO 200 II=2,NRBAY
330          NPI = NPINTS(II+1)
331          READ (5,1002) (XO(I,K),K=1,NPI)
332          Y = RB(II)
333          XB = R = SQRT(R**2-Y**2)
334          W = 0.0
335          W = AI(1) + AI(2)*Y + AI(3)*Y**2 + AI(4)*Y**3 + AI(5)*Y**4
336          XT = R + SQRT(R**2-Y**2) - W
337          XO(I) = XB
338          XO(NPI) = XT
339      C
340      DO 120 I=1,NPI
341          THETA = 0.0

```

```

342 DO 140 J=1,NC1R
343 NDP = NDP + 1
344 XYZ(NDP,1) = X0I(I)
345 XYZ(NDP,2) = RB(I)*COS(THETA)
346 XYZ(NDP,3) = RB(I)*SIN(THETA)
347 IF (I.NE.NPINTS(I)+1) GO TO 125
348 EUL(NDP,1) = ATAN2(XYZ(NDP,3),XYZ(NDP,2)) / RADDEG
349 EUL(NDP,2) = 0.0
350 IF (J .GT. 1) GO TO 122
351 TERM = (R**2 - XYZ(NDP,2)**2)
352 IF (ABS(TERM) .LT. EPS) DEG = -90.0
353 IF (ABS(TERM) .LT. EPS) GO TO 122
354 TERM1 = (1.0/2.0) * (R**2 - XYZ(NDP,2)**2) ** (-0.5) +
355 * (-2.0*XYZ(NDP,2))
356 * = AI(2) + 2.0*AI(3)*XYZ(NDP,2) + 3.0*AI(4)*XYZ(NDP,2)**2
357 * + 4.0*AI(5)*XYZ(NDP,2)**3
358 DEG = ATAN(TERM1) / RADDEG
359 122 CONTINUE
360 EUL(NDP,3) = -90.0 - DEG
361 C
362 NDPB = NDPB + 1
363 XYZB(NDPB,1) = XYZ(NDP,1)
364 XYZB(NDPB,2) = XYZ(NDP,2)
365 XYZB(NDPB,3) = XYZ(NDP,3)
366 EULB(NDPB,1) = EUL(NDP,1)
367 EULB(NDPB,2) = EUL(NDP,2)
368 EULB(NDPB,3) = EUL(NDP,3)
369 125 CONTINUE
370 IF (I.NE.3) GO TO 126
371 EUL(NDP,1) = ATAN2(XYZ(NDP,3),XYZ(NDP,2)) / RADDEG
372 EUL(NDP,3) = -ASIN((2.0*(XYZ(NDP,1)-R)) /
373 1 (4.0*(XYZ(NDP,1)-R)**2 + 4.0*XYZ(NDP,3)**2 +
374 2 4.0*XYZ(NDP,2)**2)**0.5) / RADDEG
375 IF (J .GT. NQ2) GO TO 139
376 IF (J.EQ.1 .OR. J.EQ.NQ2) GO TO 136
377 ND = ND + 1
378 JDOF(NDP,1) = ND
379 ND = ND + 1
380 JDOF(NDP,3) = ND
381 GO TO 139
382 136 CONTINUE
383 IF (J .EQ. 1) KD = 1
384 IF (J .EQ. NQ2) KD = 3
385 ND = ND + 1
386 JDOF(NDP,KD) = ND
387 GO TO 139
388 126 CONTINUE
389 IF (J .GT. NQ2) GO TO 139
390 IF (J.EQ.1 .OR. J.EQ.NQ2) GO TO 135
391 ND = ND + 1
392 JDOF(NDP,1) = ND
393 ND = ND + 1
394 JDOF(NDP,2) = ND
395 ND = ND + 1
396 JDOF(NDP,3) = ND
397 GO TO 139
398 135 CONTINUE

```

```

399      IF (J .NE. 1) GO TO 137
400      ND = ND + 1
401      JDOF(NDP,1) = ND
402      ND = ND + 1
403      JDOF(NDP,2) = ND
404      GO TO 139
405 137 CONTINUE
406      IF (I .NE. NPI) KD = 2
407      IF (I .EQ. NPI) KD = 3
408      ND = ND + 1
409      JDOF(NDP,KD) = ND
410 139 CONTINUE
411      THETA = THETA + THETAD
412 140 CONTINUE
413      IF (I .EQ. 1 .OR. I .EQ. NPI) GO TO 1122
414      NDP = NDP - NCIR
415      N1 = NDP
416      N2 = NDP + NDV2 + 2
417 C
418      N1 = N1 + 1
419      N2 = N2 - 1
420      JDOF(N2,1) = -JDOF(N1,1)
421      JDOF(N2,2) = JDOF(N1,2)
422 C
423      DO 141 J = 1,NQ1
424      N1 = N1 + 1
425      N2 = N2 - 1
426 C
427      JDOF(N2,1) = -JDOF(N1,1)
428      JDOF(N2,2) = JDOF(N1,2)
429      JDOF(N2,3) = -JDOF(N1,3)
430 141 CONTINUE
431      GO TO 1124
432 1122 CONTINUE
433 C
434      NDP = NDP - NCIR
435      N1 = NDP
436      N2 = NDP + NDV2 + 2
437 C
438      N1 = N1 + 1
439      N2 = N2 - 1
440      JDOF(N2,1) = -JDOF(N1,1)
441      JDOF(N2,2) = -JDOF(N1,2)
442 C
443      DO 1123 J = 1,NQ1
444      N1 = N1 + 1
445      N2 = N2 - 1
446 C
447      JDOF(N2,1) = -JDOF(N1,1)
448      JDOF(N2,2) = -JDOF(N1,2)
449      JDOF(N2,3) = JDOF(N1,3)
450 1123 CONTINUE
451 C
452 1124 CONTINUE
453 C
454      N1 = N1 + 1
455      NDP = NDP + NCIR

```

ORIGINAL PAGE IS
OF POOR QUALITY

ALL PAGE IS
POOR QUALITY

```

456      120 CONTINUE
457      200 CONTINUE
458
459      C
460      CALL WRITE (XYZ,NDP,3,3HXYZ,KX)
461      CALL WRITE (EUL,NDP,3,5HEULER,KX)
462      CALL WRITE (XYZB,NDPB,3,4HXYZB,KS)
463      CALL WRITE (EULB,NDPB,3,6HEULERB,KS)
464      CALL WRITIM (JDOF,NDP,6,4HJDOF,KX)
465      C-    BRING XYZ,XYZB AND EUL,EULB TOGETHER ALSO ADD THE JDOFB.
466      C
467      DO 210 I=1,NDPB
468      NDP = NDP + I
469      DO 220 J=1,3
470      XYZ(NDPA+J) = XYZB(I,J)
471      220 EUL(NDPA,J) = EULB(I,J)
472      210 CONTINUE
473      C-    JDOF TABLE (TOP CENTER POINT)
474      NDP = NDP + 1
475      NCNR = (NPINTS (2))*NCIR + NPINTS (1)
476      ND = ND + 1
477      JDOF(NDP,1) = ND
478      C-    ALL OTHER POINTS ON THE BLADDER (ON THE FIRST BAY)
479      NCNR = NCNR - NCIR - 1
480      DO 230 I=1,NCIR
481      NCNR = NCNR + 1
482      NDP = NDP + 1
483      JDOF(NDP,2) = JDOF(NCNR,2)
484      IF (I .GT. NQ2) GO TO 230
485      IF (I.EQ.1 .OR. I.EQ.NQ2) GO TO 225
486      ND = ND + 1
487      JDOF(NDP,1) = ND
488      ND = ND + 1
489      JDOF(NDP,3) = ND
490      GO TO 230
491      225 CONTINUE
492      IF (I .EQ. 1) ND = ND + 1
493      IF (I .EQ. 1) JDOF(NDP,1) = ND
494      IF (I .EQ. NQ2) ND = ND + 1
495      IF (I .EQ. NQ2) JDOF(NDP,3) = ND
496      230 CONTINUE
497      NCNR = NCNR + 1 - NCIR
498      NDP = NDP - NCIR
499      N1 = NDP
500      N2 = NDP + NDV2 + 2
501      C
502      N1 = N1 + 1
503      N2 = N2 - 1
504      JDOF(N2,1) = -JDOF(N1,1)
505      JDOF(N2,2) = -JDOF(N1,2)
506      C
507      DO 231 I = 1,NQ1
508      N1 = N1 + 1
509      N2 = N2 - 1
510      C
511      JDOF(N2,1) = -JDOF(N1,1)
512      JDOF(N2,2) = -JDOF(N1,2)

```

```

513      JDOF(N2,3) = JDOF(N1,3)
514      231 CONTINUE
515      C
516          N1 = N1 + 1
517          NDP = NDP + NCIR
518      C
519      C- ALL OTHER BLADDER POINTS EXCEPT THE LAST ONE.
520          NRBAY1 = NRBAY - 1
521          DO 250 KK=2,NRBAY1
522              NCNR = NCNR + NPINTS (KK+1) * NCIR
523              DO 240 I=1,NCIR
524                  NCNR = NCNR + 1
525                  NDP = NDP + 1
526                  JDOF(NDP,2) = JDOF(NCNR,2)
527                  IF (I .GT. NQ2) GO TO 240
528                  IF (I.EQ.1 .OR. I.EQ. NQ2) GO TO 245
529                  ND = ND + 1
530                  JDOF(NDP,1) = ND
531                  ND = ND + 1
532                  JDOF(NDP,3) = ND
533                  GO TO 240
534      245 CONTINUE
535          IF (I .EQ. 1) ND = ND + 1
536          IF (I .EQ. 1) JDOF(NDP,1) = ND
537          IF (I .EQ. NQ2) ND = ND + 1
538          IF (I .EQ. NQ2) JDOF(NDP,3) = ND
539      240 CONTINUE
540          NCNR = NCNR - NCIR
541          NDP = NDP - NCIR
542          N1 = NDP
543          N2 = NDP + NDV2 + 2
544      C
545          N1 = N1 + 1
546          N2 = N2 - 1
547          JDOF(N2,1) = -JDOF(N1,1)
548          JDOF(N2,2) = -JDOF(N1,2)
549      C
550          DO 241 I = 1,NQ1
551          N1 = N1 + 1
552          N2 = N2 - 1
553      C
554          JDOF(N2,1) = -JDOF(N1,1)
555          JDOF(N2,2) = -JDOF(N1,2)
556          JDOF(N2,3) = JDOF(N1,3)
557      241 CONTINUE
558      C
559          NDP = NDP + NCIR
560      250 CONTINUE
561      C
562      C- TAKE CARE OF THE LAST POINT WITH NO D.O.F.
563          NDP = NDP + NCIR
564      C
565      C- FOR ROTATIONAL DOF AT THE BLADDER POINTS
566      C
567          NDP = NDP - NDPRB
568          NDP = NDP + 1
569          DO 260 KK=1,NRBAY1

```

```

570      DO 270 I=1,NQ2
571      IF (I .EQ. NQ2) GO TO 264
572      IF (I .NE. 1) GO TO 262
573      NDP = NDP + 1
574      ND = ND + 1
575      JDOF(NDP,6) = ND
576      GO TO 270
577 262 CONTINUE
578      NDP = NDP + 1
579      ND = ND + 1
580      JDOF(NDP,4) = ND
581      ND = ND + 1
582      JDOF(NDP,5) = ND
583      ND = ND + 1
584      JDOF(NDP,6) = ND
585      GO TO 270
586 264 CONTINUE
587      NDP = NDP + 1
588      ND = ND + 1
589      JDOF(NDP,4) = ND
590      ND = ND + 1
591      JDOF(NDP,5) = ND
592 270 CONTINUE
593      NDP = NDP - NQ2
594      N1 = NDP
595      N2 = NDP + NDVZ + 2
596
597      N1 = N1 + 1
598      N2 = N2 - 1
599      JDOF(N2,4) = JDOF(N1,4)
600      JDOF(N2,5) = JDOF(N1,5)
601      JDOF(N2,6) = -JDOF(N1,6)
602
603      DO 271 I = 1,NQ1
604      N1 = N1 + 1
605      N2 = N2 - 1
606
607      JDOF(N2,4) = JDOF(N1,4)
608      JDOF(N2,5) = JDOF(N1,5)
609      JDOF(N2,6) = -JDOF(N1,6)
610 271 CONTINUE
611
612      NDP = NDP + NCIR
613 260 CONTINUE
614
615
616      CALL WRITE (XYZ,NDPA,3,3HXYZ,KX)
617      CALL WRITE (EUL,NDPA,3,5SHEULER,KX)
618      CALL WRITIM (JDUF,NDPA,6,4HJDUF,KX)
619
620      CCCCCC-----=====
621
622      WRITE (NUTXYZ) NDPA,NCX,NDPA,NCJ,NDPA,NCX
623      WRITE(NUTXYZ) ((JDOF(I,J),I=1,NDPA),J=1,NCJ)
624      WRITE (NUTXYZ) (( XYZ(I,J),I=1,NDPA),J=1,NCX)
625      WRITE (NUTXYZ) (( EUL(I,J),I=1,NDPA),J=1,NCX)
626

```

```

627      C- ELEMENTS IN THE FIRST BAY
628      C PENTAS
629      NJ = 1
630      NCOUNT = NELEM(1) = 1
631      DO 310 I=1,NCOUNT
632      NJ = NJ + NCIR + 1
633      N1 = NJ
634      N2 = NJ + 1
635      N3 = NJ
636      N4 = NJ - NCIR - 1
637      N5 = NJ - NCIR
638      N6 = NJ - NCIR - 1
639      DO 320 J=1,NCIR1
640      NEL = NEL + 1
641      N2 = N2 + 1
642      N3 = N3 + 1
643      N5 = N5 + 1
644      N6 = N6 + 1
645      ICONF(NEL,1) = N1
646      ICONF(NEL,2) = N2
647      ICONF(NEL,3) = N3
648      ICONF(NEL,4) = N4
649      ICONF(NEL,5) = N5
650      ICONF(NEL,6) = N6
651      320 CONTINUE
652      310 CONTINUE
653      C TETRA (TOP)
654      NJ = NJ + NCIR + 1
655      N4 = NJ
656      N1 = NJ - NCIR - 1
657      N2 = N1
658      NB1 = NTPF + 1
659      NB2 = NB1
660      DO 330 I=1,NCIR1
661      NEL = NEL + 1
662      N2 = N2 + 1
663      N3 = N2 + 1
664      ICONF(NEL,1) = N1
665      ICONF(NEL,2) = N2
666      ICONF(NEL,3) = N3
667      ICONF(NEL,4) = N4
668      NELS = NELS + 1
669      ICONS(NELS,1) = N4
670      ICONS(NELS,2) = N2
671      ICONS(NELS,3) = N3
672      NELB = NELB + 1
673      NB2 = NB2 + 1
674      NB3 = NB2 + 1
675      ICONB(NELB,1) = NB1
676      ICONB(NELB,2) = NB2
677      ICONB(NELB,3) = NB3
678      330 CONTINUE
679      NJ = NJ - (NCIR+1)
680      NJ = NJ - (NCIR+1) * NCOUNT
681      NJ1 = NJ
682      NJ2 = NPINTS (1) + (NBINTS (2)*NCIR)
683      C

```

684 C- READ IN THE BOTTOM AND TOP ELEMENTS IDENTIFICATIONS.
 685 READ (5,3002) ELMID1,ELMID2
 686 C- BOTTOM ELEMENT OF THE SECOND BAY.
 687 READ (5,3002) ELMID1,ELMID2
 688 C
 689 C
 690 IF (ELMID1 .NE. 5HPENTA) GO TO 340
 691 N1 = NJ1
 692 N2 = NJ1 + NCIR + 1
 693 N3 = NJ2
 694 DO 350 I=1,NCIR1
 695 NEL = NEL + 1
 696 N1 = N1 + 1
 697 N2 = N2 + 1
 698 N3 = N3 + 1
 699 N4 = N1 + 1
 700 NS = N2 + 1
 701 N6 = N3 + 1
 702 ICONF(NEL,1) = N1
 703 ICONF(NEL,2) = N3
 704 ICONF(NEL,3) = N2
 705 ICONF(NEL,4) = N4
 706 ICONF(NEL,5) = N6
 707 ICONF(NEL,6) = N5
 708 350 CONTINUE
 709 NJ1 = NS
 710 NJ2 = NO
 711 GO TO 370
 712 340 CONTINUE
 713 N1 = NJ1 + NCIR + 1
 714 N3 = NJ2 + NCIR + 1
 715 N5 = NJ1
 716 N7 = NJ2 + 1
 717 DO 360 I=1,NCIR1
 718 NEL = NEL + 1
 719 N1 = N1 + 1
 720 N2 = N1 + 1
 721 N3 = N3 + 1
 722 N4 = N3 - 1
 723 N5 = N5 + 1
 724 N6 = N5 + 1
 725 N7 = N7 + 1
 726 N8 = N7 - 1
 727 ICONF(NEL,1) = N1
 728 ICONF(NEL,2) = N2
 729 ICONF(NEL,3) = N3
 730 ICONF(NEL,4) = N4
 731 ICONF(NEL,5) = N5
 732 ICONF(NEL,6) = N6
 733 ICONF(NEL,7) = N7
 734 ICONF(NEL,8) = N8
 735 360 CONTINUE
 736 NJ1 = N2 - NCIR
 737 NJ2 = N3 - NCIR
 738 370 CONTINUE
 739 C- IN BETWEEN ELEMENTS IN THE SECOND BAY (EXCLUDING TOP + BOTTOM)
 740 C

```

741      NCOUNT = NELEM(2) - 2
742      N1 = NJ1 + NCIR + 1
743      N3 = NJ2 + NCIR + 1
744      NS = NJ1
745      N7 = NJ2 + 1
746      DO 380 I=1,NCOUNT
747      DO 390 J=1,NCIR1
748      NEL = NEL + 1
749      N1 = N1 + 1
750      N2 = N1 + 1
751      N3 = N3 + 1
752      N4 = N3 - 1
753      N5 = N5 + 1
754      N6 = N5 + 1
755      N7 = N7 + 1
756      N8 = N7 - 1
757      ICONF(NEL,1) = N1
758      ICONF(NEL,2) = N2
759      ICONF(NEL,3) = N3
760      ICONF(NEL,4) = N4
761      ICONF(NEL,5) = N5
762      ICONF(NEL,6) = N6
763      ICONF(NEL,7) = N7
764      ICONF(NEL,8) = N8
765      390 CONTINUE
766      N1 = N1 - NCIR1 + NCIR + 1
767      N3 = N3 - NCIR1 + NCIR
768      N5 = N5 - NCIR1 + NCIR + 1
769      N7 = N7 - NCIR1 + NCIR
770      NJ1 = N2 - NCIR
771      NJ2 = N3 - NCIR - 1
772      380 CONTINUE
773      C
774      C- FOR THE LAST ELEMENT
775      NKOUNT = NPINTS(2)
776      IF (ELMIDI .EQ. 5HPERTA) NKOUNT = NKOUNT - 1
777      IF (NKOUNT .GT. NPINTS(3)) NJ3 = NJ1 + NCIR + 1
778      IF (NKOUNT .LT. NPINTS(3)) NJ3 = NJ2 + NCIR
779      IF (NKOUNT .EQ. NPINTS(3)) GO TO 400
780      C
781      N81 = NTPF + 1 + 1
782      N83 = NTPF + NCIR + 1
783      N1 = NJ1
784      N2 = NJ2
785      N3 = NJ3
786      DO 410 I=1,NCIR1
787      NEL = NEL + 1
788      N1 = N1 + 1
789      N2 = N2 + 1
790      N3 = N3 + 1
791      N4 = N1 + 1
792      N5 = N2 + 1
793      N6 = N3 + 1
794      ICONF(NEL,1) = N1
795      ICONF(NEL,2) = N3
796      ICONF(NEL,3) = N2
797      ICONF(NEL,4) = N4

```

```

798      ICONF(NEL,5) = N6
799      ICONF(NEL,6) = NS
800      IF (NKOUNT .LT. NPINTS(3)) GO TO 395
801      NS1 = N3
802      NS2 = N2
803      NS3 = N5
804      NS4 = N6
805      NB1 = NB1 + 1
806      NB2 = NB1 - 1
807      NB3 = NB3 + 1
808      NB4 = NB3 + 1
809      GO TO 397
810      395 CONTINUE
811      NS1 = N1
812      NS2 = N3
813      NS3 = N6
814      NS4 = N4
815      NB1 = NB1 + 1
816      NB2 = NB1 - 1
817      NB3 = NB3 + 1
818      NB4 = NB3 + 1
819      397 CONTINUE
820      NELB = NELB + 1
821      ICONB(NELB,1) = NB1
822      ICONB(NELB,2) = NB2
823      ICONB(NELB,3) = NB3
824      ICONB(NELB,4) = NB4
825      NELS = NELS + 1
826      ICONS(NELS,1) = NS1
827      ICONS(NELS,2) = NS2
828      ICONS(NELS,3) = NS3
829      ICONS(NELS,4) = NS4
830      410 CONTINUE
831      IF (NKOUNT .GT. NPINTS(3)) NJ2 = NJ2
832      IF (NKOUNT .LT. NPINTS(3)) NJ2 = NJ3
833      GO TO 430
834      400 CONTINUE
835      NJ3 = NJ2 + NCIR
836      NJ4 = NJ1 + NCIR+1
837      N1 = NJ4
838      N4 = NJ3
839      N5 = NJ1
840
841      N8 = NJ2
842      DO 420 I=1,NCIR
843      NEL = NEL + 1
844      N1 = N1 + 1
845      N2 = N1 + 1
846      N4 = N4 + 1
847      N3 = N4 + 1
848      N5 = N5 + 1
849      N6 = N5 + 1
850      N8 = N8 + 1
851      N7 = N8 + 1
852      ICONF(NEL,1) = N1
853      ICONF(NEL,2) = N2
854      ICONF(NEL,3) = N3

```

```

855      ICONF(NEL,4) = N4
856      ICONF(NEL,5) = N5
857      ICONF(NEL,6) = N6
858      ICONF(NEL,7) = N7
859      ICONF(NEL,8) = N8
860      NS1 = N2
861      NS2 = N1
862      NS3 = N4
863      NS4 = N3
864      NELS = NELS + 1
865      ICONS(NELS,1) = NS1
866      ICONS(NELS,2) = NS2
867      ICONS(NELS,3) = NS3
868      ICONS(NELS,4) = NS4
869      NELB = NELB + 1
870      NB1 = NB1 + 1
871      NB2 = NB1 + 1
872      NB3 = NB3 + 1
873      NB4 = NB3 + 1
874      ICONB(NELB,1) = NB1
875      ICONB(NELB,2) = NB2
876      ICONB(NELB,3) = NB3
877      ICONB(NELB,4) = NB4
878      420 CONTINUE
879      NJ2 = NJ3
880      NJ3 = NJ4
881      430 CONTINUE
882      C-
883      C-      ALL OTHER ELEMENTS FROM THIRD TO LAST BUT ONE BAY.
884      C-
885      NBT1 = NTPF + 1 + 1
886      NBT3 = NTPF + NCIR + 1
887      NB1 = NBT1
888      NB3 = NBT3
889      NJ1 = NJ3 + NCIR + 1
890      NJ2 = NJ2 + NCIR
891      DO 500 II=3,NRBAY1
892      NB1 = NB1 + NCIR
893      NB3 = NB3 + NCIR
894      READ (5,3002) ELMID1,ELMID2
895      C-      FIRST (BOTTOM) ELEMENT OF EACH
896      IF (ELMID1 .NE. SHPENTA) GO TO 510
897      NJJ1 = NJ1
898      NJJ2 = NJ2
899      NJJ3 = NJ1 + NCIR
900      N1 = NJJ1
901      N2 = NJJ2
902      N3 = NJJ3
903      DO 520 I=1,NCIR1
904      NEL = NEL + 1
905      N1 = N1 + 1
906      N2 = N2 + 1
907      N3 = N3 + 1
908      N4 = N1 + 1
909      N5 = N2 + 1
910      N6 = N3 + 1
911      ICONF(NEL,1) = N1

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

912      ICONF(NEL,2) = N3
913      ICONF(NEL,3) = N2
914      ICONF(NEL,4) = N4
915      ICONF(NEL,5) = N6
916      ICONF(NEL,6) = N5
917
520  CONTINUE
918      NJC1 = NJJ3
919      NJC2 = NJJ2
920      GO TO 540
921
510  CONTINUE
922      NJJ1 = NJ1
923      NJJ2 = NJ2
924      NJJ3 = NJ2 + NCIR
925      NJJ4 = NJ1 + NCIR
926      N1 = NJJ4
927      N4 = NJJ3
928      N5 = NJJ1
929      N8 = NJJ2
930      DO 530 I=1,NCIRI
931      NEL = NEL + 1
932      N1 = N1 + 1
933      N2 = N1 + 1
934      N4 = N4 + 1
935      N3 = N4 + 1
936      N5 = N5 + 1
937      N6 = N5 + 1
938      N8 = N8 + 1
939      N7 = N8 + 1
940      ICONF(NEL,1) = N1
941      ICONF(NEL,2) = N2
942      ICONF(NEL,3) = N3
943      ICONF(NEL,4) = N4
944      ICONF(NEL,5) = N5
945      ICONF(NEL,6) = N6
946      ICONF(NEL,7) = N7
947      ICONF(NEL,8) = N8
948
530  CONTINUE
949      NJC1 = NJJ4
950      NJC2 = NJJ3
951
540  CONTINUE
952
C      IF ( NELEM1(I) .LE. 2) GO TO 1050
953
C-      IN BETWEEN ELEMENTS ELEMENTS OF EACH
954      NCOUNT = NELEM1(I) - 2
955
956      DO 550 I=1,NCOUNT
957      NJC3 = NJC2 + NCIR
958      NJC4 = NJC1 + NCIR
959      N1 = NJC4
960      N4 = NJC3
961      N5 = NJC1
962      N8 = NJC2
963      DO 560 J=1,NCIRI
964      NEL = NEL + 1
965      N1 = N1 + 1
966      N2 = N1 + 1
967      N4 = N4 + 1
968      N3 = N4 + 1

```

```

969      N5 = N5 + 1
970      N6 = N5 + 1
971      N8 = N8 + 1
972      N7 = N8 + 1
973      ICONF(NEL,1) = N1
974      ICONF(NEL,2) = N2
975      ICONF(NEL,3) = N3
976      ICONF(NEL,4) = N4
977      ICONF(NEL,5) = N5
978      ICONF(NEL,6) = N6
979      ICONF(NEL,7) = N7
980      ICONF(NEL,8) = N8
981      560 CONTINUE
982      NJC1 = NJC4
983      NJC2 = NJC3
984      550 CONTINUE
985
986      C
987      1050 CONTINUE
988      C= IF ( NELEM(I) .LT. 2) GO TO 580
989      C= FOR THE LAST ELEMENT IN EACH BAY.
990      C
991      NKOUNT = NPINTS(I)
992      IF (ELMIDI .EQ. SHPENTA) NKOUNT = NKOUNT - 1
993      IF (NKOUNT .GT. NPINTS (I+1)) NJC3 = NJC1 + NCIR
994      IF (NKOUNT .LT. NPINTS (I+1)) NJC3 = NJC2 + NCIR
995      IF (NKOUNT .EQ. NPINTS (I+1)) GO TO 565
996      N1 = NJC1
997      N2 = NJC2
998      N3 = NJC3
999      DO 570 I=1,NCIR
1000      NEL = NEL + 1
1001      N1 = N1 + 1
1002      N2 = N2 + 1
1003      N3 = N3 + 1
1004      N4 = N1 + 1
1005      N5 = N2 + 1
1006      N6 = N3 + 1
1007      ICONF(NEL,1) = N1
1008      ICONF(NEL,2) = N3
1009      ICONF(NEL,3) = N2
1010      ICONF(NEL,4) = N4
1011      ICONF(NEL,5) = N6
1012      ICONF(NEL,6) = N5
1013      IF (NKOUNT .LT. NPINTS (I+1)) GO TO 555
1014      NS1 = N3
1015      NS2 = N2
1016      NS3 = N5
1017      NS4 = N6
1018      GO TO 557
1019      555 CONTINUE
1020      NS1 = N1
1021      NS2 = N3
1022      NS3 = N6
1023      NS4 = N4
1024      557 CONTINUE
1025      NELS = NELS + 1
1026      ICONS(NELS,1) = NS1

```

```

1026    ICONS(NELS,2) = NS2
1027    ICONS(NELS,3) = NS3
1028    ICONS(NELS,4) = NS4
1029    NELB = NELB + 1
1030    NB1 = NB1 + 1
1031    NB2 = NB1 - 1
1032    NB3 = NB3 + 1
1033    NB4 = NB3 + 1
1034    ICONB(NELB,1) = NB1
1035    ICONB(NELB,2) = NB2
1036    ICONB(NELB,3) = NB3
1037    ICONB(NELB,4) = NB4
1038    570 CONTINUE
1039    IF (NKOUNT .GT. NPINTS (II+1)) NJC1 = NJC3
1040    IF (NKOUNT .LT. NPINTS (II+1)) NJC2 = NJC3
1041    GO TO 580
1042    565 CONTINUE
1043    NJC3 = NJC2 + NCIR
1044    NJC4 = NJC1 + NCIR
1045    N1 = NJC4
1046    N4 = NJC3
1047    N5 = NJC1
1048    N8 = NJC2
1049    DO 590 I=1,NCIR1
1050    NEL = NEL + 1
1051    N1 = N1 + 1
1052    N2 = N1 + 1
1053    N4 = N4 + 1
1054    N3 = N4 + 1
1055    N5 = N5 + 1
1056    N6 = N5 + 1
1057    N8 = N8 + 1
1058    N7 = N8 + 1
1059    ICONF(NEL,1) = N1
1060    ICONF(NEL,2) = N2
1061    ICONF(NEL,3) = N3
1062    ICONF(NEL,4) = N4
1063    ICONF(NEL,5) = N5
1064    ICONF(NEL,6) = N6
1065    ICONF(NEL,7) = N7
1066    ICONF(NEL,8) = N8
1067    NELS = NELS + 1
1068    NS1 = N1
1069    NS2 = N4
1070    NS3 = N4 + 1
1071    NS4 = N1 + 1
1072    ICONS(NELS,1) = NS1
1073    ICONS(NELS,2) = NS2
1074    ICONS(NELS,3) = NS3
1075    ICONS(NELS,4) = NS4
1076    NELB = NELB + 1
1077    NB1 = NB1 + 1
1078    NB2 = NB1 - 1
1079    NB3 = NB3 + 1
1080    NB4 = NB3 + 1
1081    ICONB(NELB,1) = NB1
1082    ICONB(NELB,2) = NB2

```

```

1083      ICONB(NELB,3) = NB3
1084      ICONB(NELB,4) = NB4
1085      590 CONTINUE
1086      NJC1 = NJC4
1087      NJC2 = NJC3
1088      580 CONTINUE
1089      NB1 = NB1 - NCIR1
1090      NB3 = NB3 - NCIR1
1091      NJ1 = NJ1 + NCIR*NPINTS (II)
1092      NJ2 = NJ2 + NCIR*NPINTS (II+1)
1093      500 CONTINUE
1094      C- FOR THE LAST (ONE ELEMENT) BAY.
1095      NJ3 = NJ2- NCIR
1096      N1 = NJ1
1097      N2 = NJ2
1098      N3 = NJ3
1099      NB1 = NB1 + NCIR
1100      NB3 = NB3 + NCIR
1101      DO 610 I=1,NCIR1
1102      NEL = NEL + 1
1103      N1 = N1 + 1
1104      N2 = N2 + 1
1105      N3 = N3 + 1
1106      N4 = N1 + 1
1107      N5 = N2 + 1
1108      N6 = N3 + 1
1109      ICONF(NEL,1) = N1
1110      ICONF(NEL,2) = N3
1111      ICONF(NEL,3) = N2
1112      ICONF(NEL,4) = N4
1113      ICONF(NEL,5) = N6
1114      ICONF(NEL,6) = N5
1115      NELS = NELS + 1
1116      NS1 = N3
1117      NS2 = N2
1118      NS3 = N5
1119      NS4 = N6
1120      ICONS(NELS,1) = NS1
1121      ICONS(NELS,2) = NS2
1122      ICONS(NELS,3) = NS3
1123      ICONS(NELS,4) = NS4
1124      NELB = NELB + 1
1125      NB1 = NB1 + 1
1126      NB2 = NB1 - 1
1127      NB3 = NB3 + 1
1128      NB4 = NB3 + 1
1129      ICONB(NELB,1) = NB1
1130      ICONB(NELB,2) = NB2
1131      ICONB(NELB,3) = NB3
1132      ICONB(NELB,4) = NB4
1133      610 CONTINUE
1134      CALL WRITIM (ICONF,NEL,8,4HJFLD,1500)
1135      C-----C
1136      C
1137      C- WRITE FLUID PROPERTY ON TAPE (INTERIOR)
1138      WRITE (NUTEL,3001) AN6(2)
1139      WRITE (NUTEL,3002) AN6(7),AN6(9),AN6(4),AN6(11),AN6(4)

```

```

1140      WRITE (NUTEL,3003) ANS(1),RHOF,ANS(2),BLKM
1141      C-    WRITE FLUID ELEMENTS ON TAPE
1142          DO 405 L=1,NEL
1143          405 WRITE (NUTEL,3006) L, (ICONF(L,J),J=1,8)
1144              WRITE (NUTEL,3006) (INTG(J),J=1,2)
1145      C-
1146      C-----+
1147      C
1148      C-    WRITE PROPERTY OF GRAVITY ELEMENT ON TAPE
1149      C
1150          CALL WRITIM (ICONS,NELS,4,4HJSUR,KS)
1151          WRITE (NUTEL,3001) AN6(3)
1152          WRITE (NUTEL,3002) AN6(4),AN6(9),AN6(4),AN6(4),AN6(4)
1153          WRITE (NUTEL,3003) ANS(1),RHUF
1154          WRITE (NUTEL,3003) ANS(3),GVX,ANS(4),GVY,ANS(5),GVZ
1155      C-    WRITE GRAVITY ELEMENTS ON TAPE
1156          DO 710 LI = 1,NELS
1157              L = LI + NEL
1158              710 WRITE (NUTEL,3006) L, (ICONS(L,J),J=1,4)
1159              WRITE (NUTEL,3006) (INTG(J),J=1,2)
1160      C
1161      CCCCCC-----+
1162      C
1163      C-    WRITE PROPERTY OF TRIANGULAR BLADDER ELEMENT (NCIRI) ON TAPE
1164      C
1165          CALL WRITIM (ICONB,NELB,4,4HJBBLU,KS)
1166          WRITE (NUTEL,3001) AN6(5)
1167          WRITE (NUTEL,3002) AN6(8),AN6(9),AN6(4),AN6(4),AN6(4)
1168          WRITE (NUTEL,3003) ANS(1),RHOBBLU,ANS(6),EBLU,ANS(7),ANUBLU
1169          WRITE (NUTEL,3003) ANS(8),TBLD,ANS(9),TBLD,ANS(10),TBLD
1170      C-    WRITE BLADDER TRIANGULAR ELEMENT ON TAPE
1171          DO 725 LI = 1,NCIRI
1172              L = LI + NEL + NELS
1173              725 WRITE (NUTEL,3004) L, (ICONB(L,J),J=1,3), TBLD,TBLD,TBLD
1174              WRITE (NUTEL,3006) (INTG(J),J=1,2)
1175              IF (NELB .EQ. NCIRI) GO TO 727
1176      C
1177      C-    WRITE BLADDER QUAD. PROPERTY ON TAPE
1178          WRITE (NUTEL,3001) AN6(6)
1179          WRITE (NUTEL,3002) AN6(8),AN6(9),AN6(4),AN6(4),AN6(4)
1180          WRITE (NUTEL,3003) ANS(1),RHOBBLU,ANS(6),EBLU,ANS(7),ANUBLU
1181          WRITE (NUTEL,3003) ANS(8),TBLD,ANS(9),TBLD,ANS(10),TBLD
1182      C-    WRITE BLADDER QUAD. ELEMENTS ON TAPE
1183          DO 728 LI = NCIRI,NELB
1184              L = LI + NEL + NELS
1185              728 WRITE (NUTEL,3005) L, (ICONB(L,J),J=1,4), TBLD,TBLD,TBLD
1186              WRITE (NUTEL,3006) (INTG(J),J=1,2)
1187              727 CONTINUE
1188                  WRITE (NUTEL,3001) AN6(1)
1189      C
1190      C-----+
1191      C
1192          RETURN
1193          END

```

Appendix - B1EXPLANATION OF
INPUT TO STATIC FREE SURFACE PROGRAM

Card Nos.	Input	Explanation
1	Run no., cols. 1-6; name, cols. 11-28	Three cards to satisfy subroutine "START"
2	Title 1, cols. 1-78	
3	Title 2, cols. 1-78	
4	Order of the polynomial, acceleration due to gravity (in/sec ²), ullage pressure (psi), ratio of volume fill to total volume	Format (I5,3D17.8)
5	'STOP', cols. 1-4	End of data

1 - SEPS60 SINGH
2 ---- FREE SURFACE FOR MERCURY SLOSH-----60 PERCENT FULL
3 ---- GRAVITY FIELD-----0.00386
4 ----- 4 0.00386 0.10 0.60
5 ---- STOP

ORIGINAL
OF POOR QUALITY

Appendix - B2EXPLANATION OF
INPUT FOR DYNAMIC ANALYSIS PROGRAM

Card Nos.	Input	Explanation
1-31		Control cards (if needed)
32	Run no., cols. 1-6; name, cols. 11-28	
33	Title 1, cols. 1-78	
34	Title 2, cols. 1-78	}
35	Users comment cards	Three cards to satisfy subroutine "START"
36	'INITIL' or 'NOINIT', cols. 1-6	Any no. of cards, the last card must be zeros. Cols. 1-10. Subroutine "COMENT"
37	'GNXYZ', cols. 1-5	To initialize or not to initialize the reserve tape
38	'XYZEUL', cols. 1-6	To call subroutine "GNXYZ3"
39	Radius of the sphere, cols. 1-10	To call subroutine "XYZEUL"
40	2 x mass density, 2 x bulk modulus, of the fluid and gravity components in x, y, and z directions	Format (F10.0)
41	2 x mass density, 2 x Young's modulus, Poisson's ratio, and thickness of the bladder	Format (5F10.0)
42	Number of circumferential points and number of radial bays in the half space model	Format (4F10.0)
43	Name, no. of rows, no. of cols. of the matrix	Format (2I5)
44	The coefficients of the polynomial assumed for the free surface	Subroutine "READ"
45		
46	Zeros	
47	Name, no. of rows, no. of cols. of the matrix	Subroutine "READ"
48		
49	bay radii	
50		
51	Zeros	

ORIGINAL PAGE IS
OF POOR QUALITY

52	Name, no. of rows, no. of cols. of the matrix	Subroutine "READIM"
53	Number of points on each vertical line	
54	Zeros	
58-70	X-coord. of the points on each vertical line from inside-out	Format (8F10.0)
71-81	Identification of the first and the last element in each bay from bottom to top	Format (2(A6,4X))
82	'FINEL', cols. 1-5	Calls subroutine "FINEL"
83	'MODES', cols. 1-5	Calls subroutine "OYMODE"
84	Name, no. of rows, no. of cols. of the matrix	Matrix of assumed mode shapes (if any). Subroutine "READ"
85	Zeros	
86	Number of modes wanted	Format (10X,I5)
87	Number of modes used	Format (10X,I5)
88	Shift value for ω^2 (convergence will be about this value)	Format (10X,E17.0)
89	No. of maximum iteration allowed	Format (10X,I5)
90	'PUNCH' or 'NOPNCH', cols. 1-6	Option for punch output
91	'MECHEQ', cols. 1-6	Calls subroutine "MECHEQ" to calculate mechanical equivalent
92	Name, logical tape no., run no.	To read x, y, z locations of the points. Subroutine "READ"
93	Last joint with only 3d.o.f., last d.o.f. number	Format (2I5)
94	Name, no. of rows, no. of cols. in the matrix	Location of the reference point. Subroutine "READ"
95	Zeros	
96	Name, no. of rows, no. of cols. in the matrix	The elements of the matrix indicate which cols. of the rigid body matrix are non-zeros. Subroutine "READIM"
97	Column no. of rigid body matrix (non-zero)	
98	Zeros	
99	Name, logical tape no., run no.	Degrees of freedom matrix. Subroutine "READ"

100	Name, logical tape no., run no.	Euler angle matrix Subroutine "READ"
101	Name, logical tape no., run no.	Mass matrix Subroutine "READ"
102	Name, logical tape no., run no.	Modes matrix Subroutine "READ"
103	'PLOT', cols. 1-4	Option to plot the mode shapes (NO card, no plot)
104	Name, no. of rows, no. of cols. in the matrix	Joint numbers which are to be plotted Subroutine "READIM"
105	Node no. to be plotted	
106	Zeros	
107	Name, no. of rows, no. of cols. of the matrix	x, y, z location of the points Subroutine "READ"
108	Name, no. of rows, no. of cols. of the matrix	Euler angles of the points Subroutine "READ"
109	Name, no. of rows, no. of cols. of the matrix	Frequencies of the system Subroutine "READ"
110	Name, no. of rows, no. of cols. of the matrix	Modes of the system Subroutine "READ"
111	Name, no. of rows, no. of cols. of the matrix	d.o.f. matrix of the system Subroutine "READ"
112	'START', cols. 1-5	Preparatory to end
113	'STOP', cols. 1-4	End of data
114-118		Control cards

1
2 =RUN,Q M08GB1,HMMHEA8704,S-BULTM00000,30,2000/1200 BULTMAN BIN S-190
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

=US*ER.BOXPCN ,S-195-BULT.
=FREE TPF\$.
=ASG,T TPF\$,F2/1/PDS/500
=ASG,T T,U,17701
=COPY,G T.,TPF\$.
=FREE T.
=ASG,T 1.,F40/1/POS/10
=ASG,T 2.,F40/1/POS/10
=ASG,T 11.,F40/1/POS/10
=ASG,T 12.,F40/1/POS/10
=ASG,T 13.,F40/1/POS/10
=ASG,T 14.,F40/1/POS/10
=ASG,T 15.,F40/1/POS/10
=ASG,T 16.,F40/1/POS/10
=ASG,T 17.,F40/1/POS/10
=ASG,T 21.,F40/1/POS/10
=ASG,T 22.,F40/1/POS/10
=ASG,T 23.,F40/1/POS/10
=ASG,T 24.,F40/1/POS/10
=ASG,T 25.,F40/1/POS/10
=ASG,T 26.,F40/1/POS/10
=ASG,T 27.,F40/1/POS/10
=ASG,T 28.U
=ASG,T 30.,F40/1/POS/10
=ASG,T 31.,F40/1/POS/10
=ELT,TLD DATA
RUN140 PHILIPPUS

LATERAL SLOSH FOR SPHERICAL TANK WITH MERCURY AND BLADDER IN IT.
SPHERE RIGID----- 40 PERCENT FULL----- G # 386.0 IN / SEC**2.
0000000000

ORIGINAL PAGE IS
OF POOR QUALITY

INITIAL

GNXYZ

XYZEUL

8.00

	0.0026	2000000.0	-386.0	0.0	0.0	0
	0.000228	400.00	0.45	0.06		
	7	12				
	AI	1	5			
	1	1	8.426723	0.00	1.164276	-0.356903
	1	5	0.024364			
	0000000000					
	RR	1	12			
	1	1	1.00	2.00	2.75	3.30
	1	5	4.30	5.40	6.20	6.9
	1	9	7.3	7.5	7.9	8.0
	0000000000					
	NPTS	1	13			
	1	1	5	4	4	2
	0000000000					
	NELEM1	1	12			
	1	1	4	3	3	1
	0000000000					

58 0.0 2.3 4.1 6.68 7.57
 59 0.06275 2.3 4.1 6.68
 60 0.3 2.3 4.1 5.13
 61 0.5 2.3 4.1 4.28
 62 0.71 2.3 4.1
 63 1.3 2.3 4.1 4.75
 64 2.3 4.1 4.75 7.0
 65 2.95 4.1 4.75 7.0 9.0
 66 4.1 4.75 7.0 9.0 10.2
 67 4.75 7.0 9.0 10.2 10.5
 68 5.4 7.0 9.0 10.2
 69 7.0 9.0
 70 8.00
 71 PENTA TETRA
 72 HEXA HEXA
 73 HEXA HEXA
 74 HEXA PENTA
 75 HFXA PENTA
 76 PENTA PENTA
 77 HFXA PENTA
 78 PENTA PENTA
 79 PENTA PENTA
 80 HEXA PENTA
 81 PENTA PENTA
 82 FINFL
 83 MODES
 84 INMODE 1 1
 85 0000000000
 86 NW 6
 87 NU 6
 88 SHIFT 500.0
 89 MAXIT 15
 90 NOPUNCH
 91 MECHED
 92 XYZ -28RUN140
 93 321 375
 94 XREF 1 3
 95 0000000000
 96 JVEC 1 6
 97 1 1 1 2 0 0 0 3
 98 0000000000
 99 JDOF -28RUN140
 100 EUL -28RUN140
 101 MASS -28RUN140
 102 MODES -28RUN140
 103 PLDT
 104 SURFAS 1 13
 105 1 1 33 26 55 83 104 132 160 195 230 265 293 307 314
 106 0000000000
 107 XYZ -28RUN140
 108 EUL -28RUN140
 109 FREQ -28RUN140
 110 MODES -28RUN140
 111 JDOF -28RUN140
 112 START
 113 STOP
 114 =END
 115 =XQT MM200G
 116 =ADD,PL DATA
 117 =PMD,PLE

ORIGINAL PAGE IS
OF POOR QUALITY

B2-6

118

=FIN